# Lab 08

For this homework, provide a single rendered R Markdown file in *pdf* format on crowdmark for the problems (you may render the R Markdown file to *html*, and then convert the *html* file to *pdf* using the print function on your web browser). Indicate your student number on the markdown file before the first section header, and make a section for each lettered part of each problem (i.e., '`# Problem 1a)`', '`# Problem 1b)`', '`# Problem 2a)`' etc.). **Provide in the markdown *the final version of all of the code you wrote for this homework***, and make sure long lines of code are wrapped in the rendered *pdf*. If your Markdown file involves examining a large dataset, do not print the entire dataset to the markdown file in the steps of your solution (instead, suppress the output, or only show a small section of the data as an example). If you do the bonus problem, provide it as a separate R Markdown file in *pdf* format on crowdmark.

## Problem 1: URL extract

a) Consider the HTML file *cbc.ca.2023.03.23.html* included in the archive for this lab. That file was created by accessing the website *www.cbc.ca* on the date 23rd of March 2023, and saving the source of that website as an HTML file. Write R code to read this file into a string, and then extract every URL from that string, and store those URLs into a character vector in R called *urls*. A full description of what a URL can look like is available here: `https://www.ietf.org/rfc/rfc1738.txt`. This full description of what a URL can look like is published by the Network Working Group, established in 1972. However, for this question, you don't need to match this formal definition of URL. Instead, please match only strings starting with *http://* or *https://* followed by an uninterrupted sequence of characters, until just before we see whitespace or an end to a quoted string that was quoting the

URL. For example, in this file if we see `<link rel="preconnect" href="https://i.cbc.ca" />`, we will report *https://i.cbc.ca* as an element of the resulting character vector (without the trailing `"`). We may also see `https: backslash u002F backslash u002F www.cbc.ca backslash u002F` in this file (where by `backlash` we have the literal backslash). For this case, where a URL is encoded through another layer of abstraction that is not immediately indicated by the definition of URL, we do not report this as an element of the character vector *urls*.

The task is summarized as follows: Form a character vector *urls* in R with every substring from the file *cbc.ca.2023.03.23.html* that starts with *https://* or *http://* and ends in a single or double quote, or whitespace (right exclusive).

(9 points)

b) Report the length of the character vector *urls*.

(1 point)