MTH208: Worksheet 6

Web Scraping

In this worksheet, we will learn how to scrape the web. Scraping the web means that we can collect data that is online in a way that it can be stored in an organized way. We will need `R` package `rvest` for the scraping. Further, we will also learn some data handling with the combination of packages called `tidyverse`. (These packages have already been installed on your machines.)

```
library(tidyverse)
library(rvest)
```

Our objective in the first task is to scrape the names of all the faculty in the Department of Mathematics and Statistics at IIT Kanpur.

Function `read_html` goes to a website and reads and saves the html code of that website.

```
html <- read_html("https://www.iitk.ac.in/math/faculty")
```

The complete html code for the website is saved in the object `html`. You may go to the website, view source code and see what the html code looks like.

Since the goal is to extract all faculty's name, we go to the source code to find the first faculty's name: "ANAND A.":

```
<tr>
<td colspan="2">
<h3 class="head3"><a href="/math/../new/akash-anand" target="_blank"
style="font-size: 15px !important; font-weight: bold;">ANAND A.</a>
( PhD, University of Minnesota)</h3>
</td>
</tr>
```

We see that it looks like all faculty's names are written in the `h3` tag with class being `head3`, and further in the hyperlink tag `a`. We can use function `html_elements()` to extract all instances of a particular kind of tag.

```
# extracting all tags with class = head3. The
# "." indicates class.
name <- html_elements(html, ".head3")
```

```r
# From all the head3 class, extracting all link tags
name <- html_elements(name, "a")

# Extracting the text associated with the links
name <- html_text(name)

## A faster way
name <- html_elements(html, ".head3 a")
name <- html_text(name)
```

And we're done! Notice how we have to keep reassigning the value of `name` or find other variable names. This can be overcome by using "pipes" available in `tidyverse`. Using pipe `%>%` allows us to do a series of operations in one go. A pipe reads like "and then do.." The line below gives the same result as before!

```r
name <- html %>% html_elements(".head3 a") %>% html_text()
```

Having learned this, let's try a few tasks:

1. Write an R program to obtain the list of post doctoral fellows in the Department of Mathematics and Statistics at IIT Kanpur.

2. Write an R code to obtain the complete list of the top 250 movies on IMDb.

   ```r
   html <- read_html("https://www.imdb.com/chart/top/")
   ```

   You may not be able to get a clean list of the movie names. Using functions `strsplit`, `substring`, `gsub`, (or any other methods) create a vector of a clean list of all the movies.

3. Write an R code to obtain the following information about the top 250 IMDb movies (and store the information in a clean data.frame)

   a. Movie name

   b. Movie year

   c. Movie rating

   d. Number of votes