Statistical Computing with R Masters in Data Science 503 (S7) Fourth Batch, SMS, TU, 2025

Shital Bhandary
Associate Professor
Statistics/Bio-statistics, Demography and Public Health Informatics
Patan Academy of Health Sciences, Lalitpur, Nepal
Faculty, Masters in Medical Research, NHRC/Kathmandu University
Faculty, FAIMER Fellowship in Health Professions Education, India/USA

Review Preview (Unit 2, Part 1)

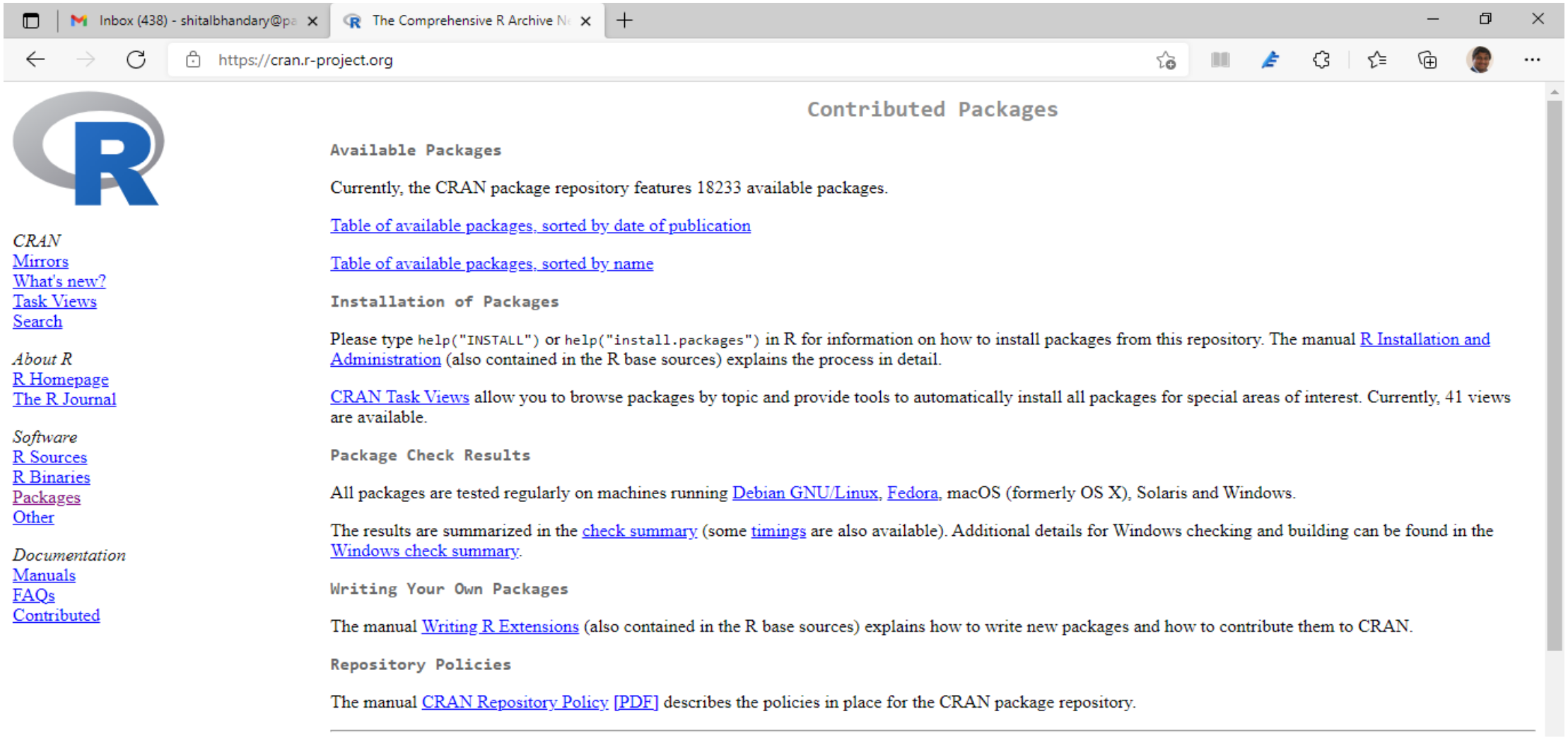
- Package in R
- Using package in R
- Reading data in R
- Reviewing data in R
- Cleaning data in R

What is a “package” in R?

- In R, the fundamental unit of shareable code/function is the package.
- A package bundles together code, data, documentation, and tests, and is easy to share with others.
- As of June 2019, there were over 14,000 packages available on the **C**omprehensive **R** **A**rchive **N**etwork, or CRAN, the public clearing house for R packages.
- This huge variety of packages is one of the reasons that R is so successful: the chances are that someone has already solved a problem that you’re working on, and you can benefit from their work by downloading their package.

Link: https://r-pkgs.org/intro.html from R-packages book: https://r-pkgs.org/index.html

Available “packages” from CRAN:



“Packages” details at CRAN:

Related Directories

Archive
Previous versions of the packages listed above, and other packages formerly available.
Orphaned
Packages with no active maintainer, see the corresponding README.
bin/windows/contrib
Windows binaries of contributed packages
bin/macosx/contrib
macOS High Sierra binaries of contributed packages
bin/macosx/el-capitan/contrib
OS X El Capitan binaries of contributed packages

How to install, use and get help about any package from CRAN?

- We can install packages of CRAN in R using:

  install.packages(“packagename”)

  e.g. install.packages(“dplyr”)
- We can then use the installed packages in R using:

  library(packagename)

  i.e. library(dplyr)
- We can get help on the installed packages in R using:

  ?packagement or help(package = “packagement”)

  e.g. ?dplyr OR help(package = “dplyr”)

“Packages” from Bioconductor repository: To work with Bioinformatics!

ot secure | www.bioconductor.org

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Search:

Home | Install | Help | Developers | About

About *Bioconductor*

Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data. *Bioconductor* uses the R statistical programming language, and is open source and open development. It has two releases each year, and an active user community. *Bioconductor* is also available as an AMI (Amazon Machine Image) and Docker images.

News

- *Bioconductor* 3.14 release schedule announced. Please view for important deadlines.
- *Bioconductor* Bioc 3.13 Released.
- *Bioconductor* browsable code base now available.
- See our google calendar for events, conferences, meetings, forums, etc. Add your event with email to events at bioconductor.org.
- *Bioconductor* F1000 Research Channel is available.
- Orchestrating single-cell analysis with *Bioconductor* (abstract; website) and other recent literature

Install »

- Discover 2042 software packages available in *Bioconductor* release 3.13.

Get started with *Bioconductor*

- Install *Bioconductor*
- Get support
- Latest newsletter
- Follow us on twitter
- Install R

Learn »

Master *Bioconductor* tools

- Courses
- Support site
- Package vignettes
- Literature citations
- Common work flows
- FAQ
- Community resources
- Videos

Use »

Create bioinformatic solutions with *Bioconductor*

- Software, Annotation, and Experiment packages
- Docker and Amazon machine images
- Latest release announcement
- Use *Bioconductor* in the AnVIL. See our project updates.
- Community Slack sign-up
- Support site
- Events calendar; email *events at*

Develop »

Contribute to *Bioconductor*

- Developer resources
- Use Bioc ‘devel’
- ‘Devel’ packages
- Package guidelines
- New package submission
- Git source control
- Build reports
- Browsable code base

How to develop a package in R?

https://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/

- Step 0: Packages you will need
- Step 1: Creating your package directory
- Step 2: Add functions
- Step 3: Add documentation
- Step 4: Process your documentation
- Step 5: Install
- Step 6: Make a package GitHub repo (Bonus!)
- Step 7: Infinity- Iterate

You do not need to create package in R for this course but it is required to know how to do it so that you can do it if required.

Reading (Import) data in R/R Studio:

- Text files: R base, readr etc. #Already covered in Unit 1
- Excel files: readXL, openxls etc. #Already covered in Unit 1
- SPSS, Stata, SAS files: foreign, haven etc. #Already covered in Unit 1

Reading data in R/R Studio:

- JSON files: rjason, jsonlite, RJSONIO etc.

Where, JSON = **J**ava**S**cript **O**bject **N**otation, used a lot in websites!

Creating a JSON file using a Text Editor (Notepad): https://www.tutorialspoint.com/r/r_json_files.htm

- Example of a JavaScript Object Notation (JSON) file:


```
{
  "ID":["1","2","3","4","5","6","7","8" ],
  "Name":["Rick","Dan","Michelle","Ryan","Gary","Nina","Simon","Guru" ],
  "Salary":["623.3","515.2","611","729","843.25","578","632.8","722.5" ],
"StartDate":["1/1/2012","9/23/2013","11/15/2014","5/11/2014","3/27/2015","5/21/2013","7/30/2013","6/17/2014"],
  "Dept":["IT","Operations","IT","HR","Finance","IT","Operations","Finance"]
}
```


- It can be typed in text editor and saved with .json extension e.g. jason_data.json

Read the created JSON file in R and Convert it as data.frame for further manipulation in R:

- Install.packages(“rjson”)
- library(“rjson”)
- data <- fromJSON(file = “jason_data.json”)
  # jason_data.json must be in the working directory of R!
- print(data)

- Covert to data frame:
- jason_data_frame <- as.data.frame(data)
- print(jason_data_frame)

#Get summary, histogram of salary,
Average salary by department
#Frequency distribution of all variables

Reading data in R/R Studio: Web Scrapping

- JSON files: rjason, jsonlite, RJSONIO etc.
- HTML page: rvest package (from R Studio)
- Use “rvest” package to extract HTML <table>

Reading JSON file from URL: Web API

https://www.geeksforgeeks.org/working-with-json-files-in-r-programming/

- install.packages("jsonlite") #Package "RJSONIO" also works!
- library(jsonlite)

- Raw <- fromJSON("https://data.ny.gov/api/views/9a8c-vfzj/rows.json?accessType=DOWNLOAD") #Large list!

- food_market <- Raw[['data']]
- str(food_market) #Large Matrix, 28472 rows and 24 columns!
- head(food_market)
- Names <- food_market[,14] #Large characters, Col 14 only!
- heads(Names) #Few names from Column 14!

What <u>more</u> can you do with the food_market data?

- Try: table(Names)

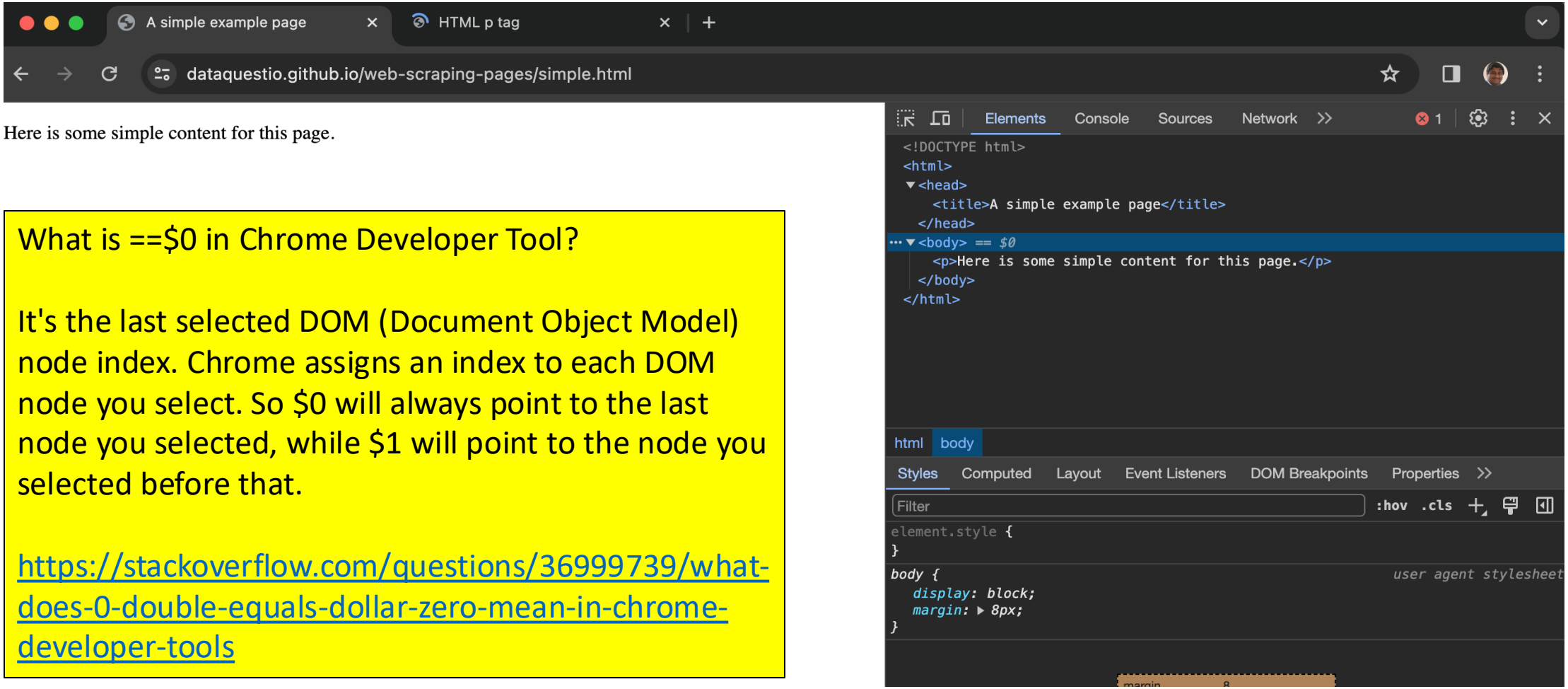
- Try: table(V19) #Why error?

- Try: table(food_market$V19) #What is “atomic vector”?
- Try: table(food_market[,19]) #What do you get?

- Convert the food_market data to data.frame and get summary, create plots of all the “useful” variables and compute appropriate averages too!

HTML scrapping: Inspect in Google Chrome

https://dataquestio.github.io/web-scraping-pages/simple.html



What is ==$0 in Chrome Developer Tool?

It's the last selected DOM (Document Object Model) node index. Chrome assigns an index to each DOM node you select. So $0 will always point to the last node you selected, while $1 will point to the node you selected before that.

https://stackoverflow.com/questions/36999739/what-does-0-double-equals-dollar-zero-mean-in-chrome-developer-tools

Web scrapping in R: A Simple (barebones) Example https://www.dataquest.io/blog/web-scraping-in-r-rvest

- The recommended package for web scrapping in R is “rvest”
- install.packages(“rvest”)
- library(rvest)
- simple <- read_html("https://dataquestio.github.io/web-scraping-pages/simple.html")
- simple %>%

  html_nodes(“p”) %>%

  html_text()

This is some text in a paragraph.

It is embedded in <body>

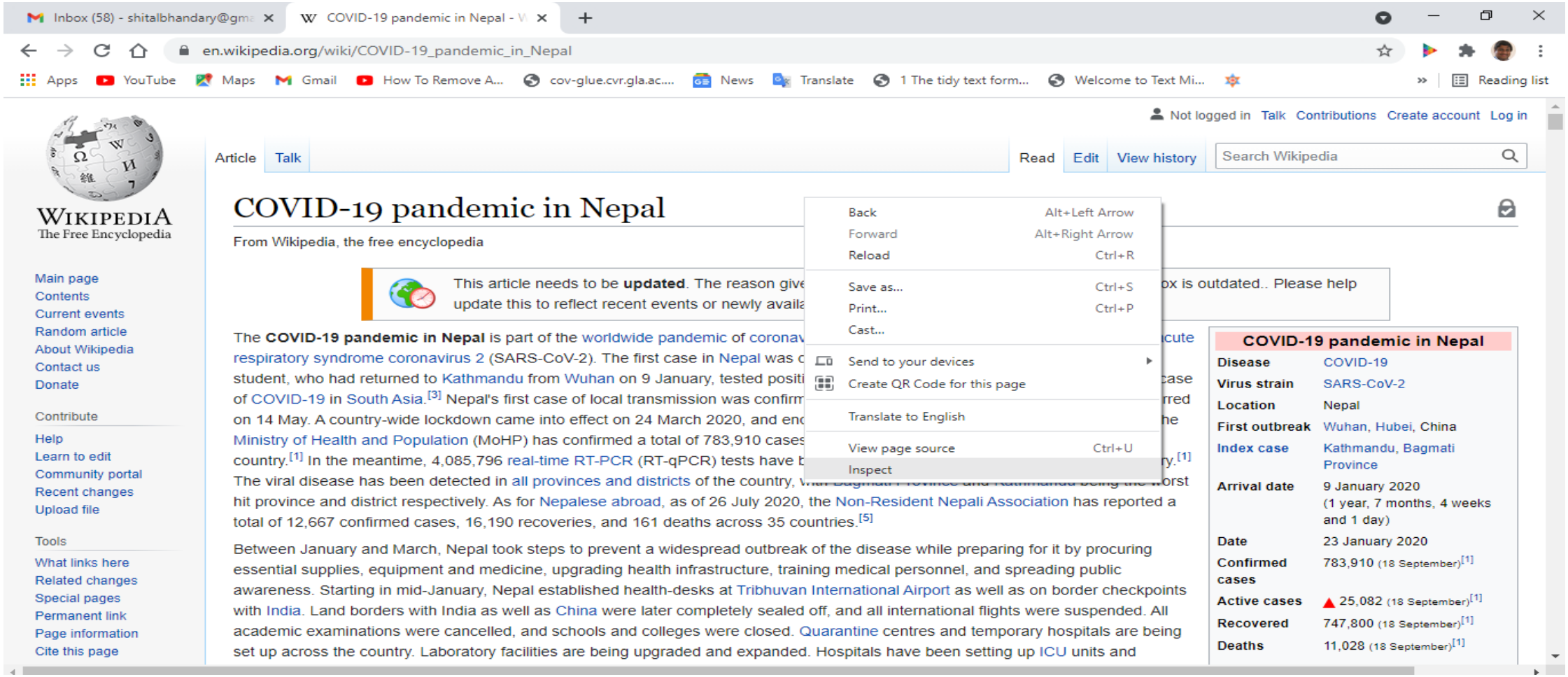
rvest package: html_node() or html_nodes() find HTML tags (nodes) using CSS selectors or XPath expressions

A simple HTML Table node with <table> and <td>: https://www.w3schools.com/html/html_tables.asp

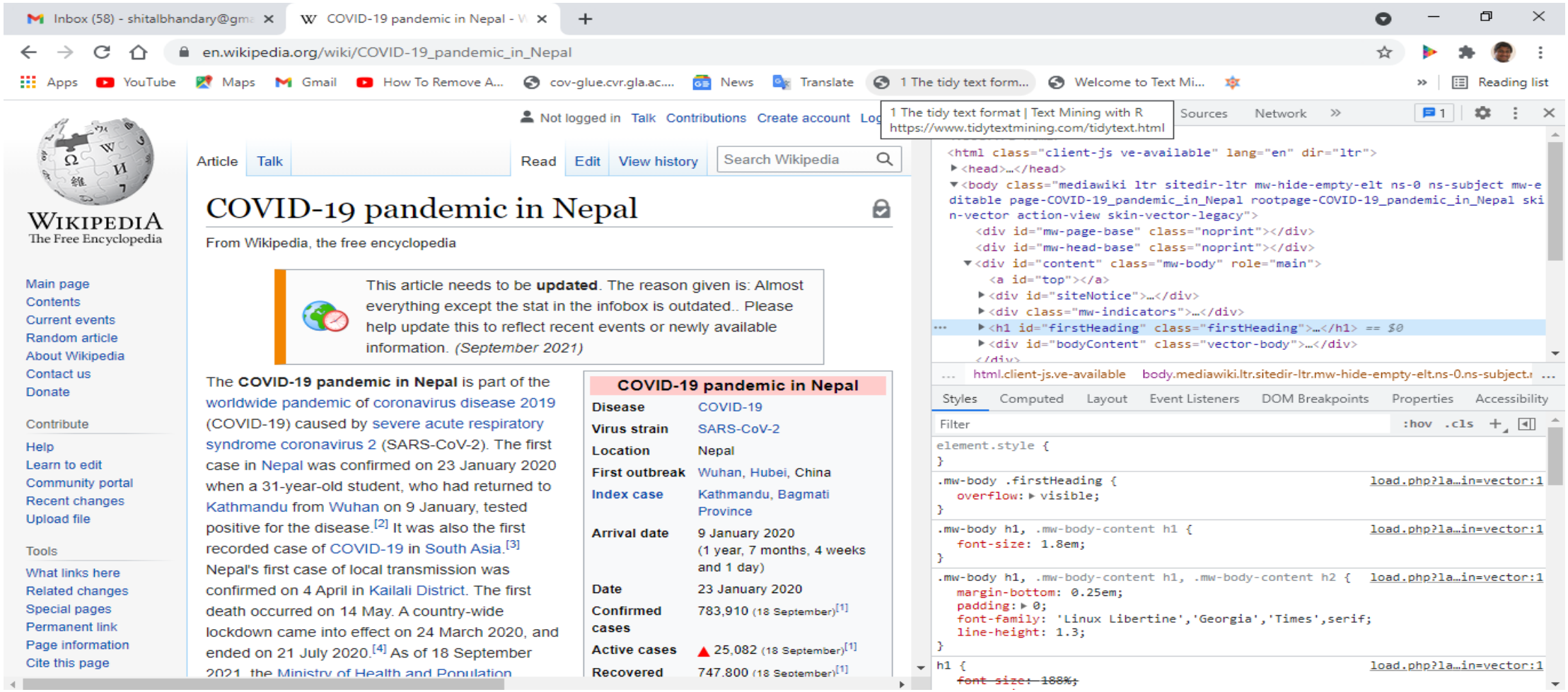
Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

- <table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>

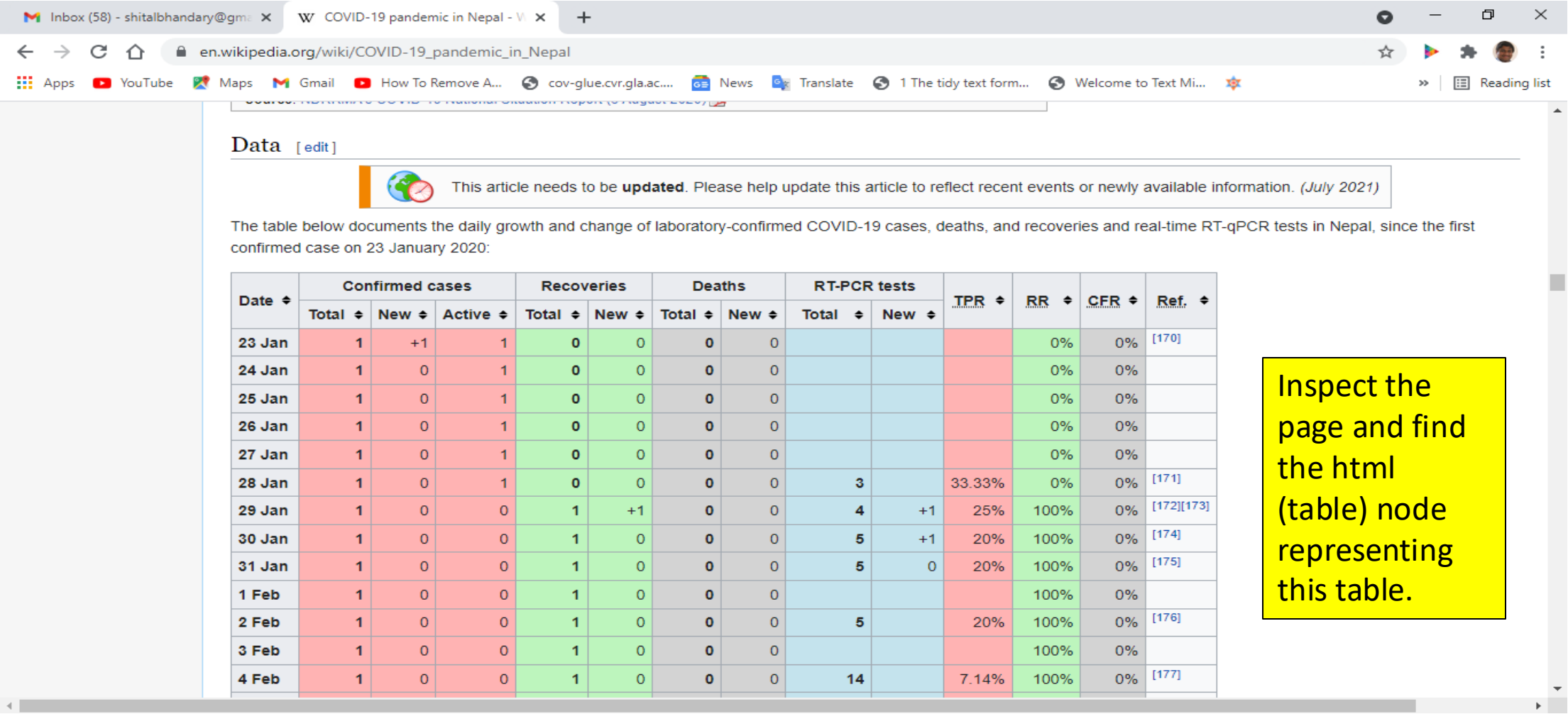
How to do web scrapping of this page? Inspect the HTML elements!



What to do now?



We/you need to scrap this data (table) in R: And create plots, get summaries etc.



Data [edit]

This article needs to be **updated**. Please help update this article to reflect recent events or newly available information. *(July 2021)*

The table below documents the daily growth and change of laboratory-confirmed COVID-19 cases, deaths, and recoveries and real-time RT-qPCR tests in Nepal, since the first confirmed case on 23 January 2020:

Date	Confirmed cases			Recoveries		Deaths		RT-PCR tests		TPR	RR	CFR	Ref.
	Total	New	Active	Total	New	Total	New	Total	New				
23 Jan	1	+1	1	0	0	0	0				0%	0%	[170]
24 Jan	1	0	1	0	0	0	0				0%	0%	
25 Jan	1	0	1	0	0	0	0				0%	0%	
26 Jan	1	0	1	0	0	0	0				0%	0%	
27 Jan	1	0	1	0	0	0	0				0%	0%	
28 Jan	1	0	1	0	0	0	0	3		33.33%	0%	0%	[171]
29 Jan	1	0	0	1	+1	0	0	4	+1	25%	100%	0%	[172][173]
30 Jan	1	0	0	1	0	0	0	5	+1	20%	100%	0%	[174]
31 Jan	1	0	0	1	0	0	0	5	0	20%	100%	0%	[175]
1 Feb	1	0	0	1	0	0	0				100%	0%	
2 Feb	1	0	0	1	0	0	0	5		20%	100%	0%	[176]
3 Feb	1	0	0	1	0	0	0				100%	0%	
4 Feb	1	0	0	1	0	0	0	14		7.14%	100%	0%	[177]

Inspect the page and find the html (table) node representing this table.

We can do as follows in R/R Studio:

- library(rvest)
- wiki_link <- "https://en.wikipedia.org/wiki/COVID-19_pandemic_in_Nepal"
- wiki_page <- read_html(wiki_link) #rvest package
- str(wiki_page)
- wiki_page %>% html_nodes("table") #to get tables with index

#Get the desired table using “div class”:

- covid_table <- wiki_page %>%
  html_element("div.COVID-19_pandemic_data_Nepal_medical_cases") %>%
  html_nodes("table") %>% html_table() %>% .[[1]] #. for getting first table!
- str(covid_table) **#tibble [496 x 14]; tibble = fast data frame!**

Data wrangling: Part I
Column names of covid_table

#Changing names of the columns by adding values of first row

- names(covid_table) <- paste(names(covid_table), covid_table[1,], sep = "_")

#Removing the first row thereafter

- covid_table <- covid_table[-1,]

#Check the structure of data again

- str(covid_table)

Data wrangling: Part II
Now do as follows in R for “covid-table” data:

- Change “Date_Date” variable as “Date”
- Change “Confirmed cases_Total” variable as “Confirmed_Cases_Total”
- Change “Confirmed cases_New” variable as “Confirmed_Cases_New”
- Change “Confirmed cases_Active” variable as “Confirmed_Cases_Active”
- Change “RT-PCR tests_Total” variable as “RT-PCR_tests_Total”
- Change “RT-PCR tests_New” variable as “RT-PCR_tests_New”
- Change “TPR_TPR” variable as “TPR”
- Change “RR_RR” variable as “RR”
- Change “CFR_CFR” variable as “CFR”
- Change “Ref. _Ref.” variable as “Ref”

Data wrangling: Part I

You can use other method too!

- Check if this works or not!

- colnames(covid_table) <- c("Date", "Confirmed_Cases_Total", "Confirmed_Cases_New", "Confirmed_Cases_Active", "Recoveries_Total", "Recoveries_New", "Deaths_Total", "Deaths_New", "PCR_Total", "PCR_New", "TPR", "RR", "CFR", "Ref")

- str(covid_table)

OR, will this also work?

```
colnames(covid_table)
names(covid_table)[names(covid_table) == "Date_Date"] = "Date"
names(covid_table)[names(covid_table) == "Confirmed cases_Total"] = "Confirmed_Cases_T
names(covid_table)[names(covid_table) == "Confirmed cases_New"] = "Confirmed_Cases_New
names(covid_table)[names(covid_table) == "Confirmed cases_Active"] = "Confirmed_Cases_
names(covid_table)[names(covid_table) == "RT-PCR tests_Total"] = "PCR_Total"
names(covid_table)[names(covid_table) == "RT-PCR tests_New"] = "PCR_New"
names(covid_table)[names(covid_table) == "TPR_TPR"] = "TPR"
names(covid_table)[names(covid_table) == "RR_RR"] = "RR"
names(covid_table)[names(covid_table) == "CFR_CFR"] = "CFR"
names(covid_table)[names(covid_table) == "Ref._Ref."] = "Ref"
colnames(covid_table)
```

Data wrangling: Part III
Removing “+” and “%” from variables:

#Removing + from four variables

- covid_table$Confirmed_Cases_New <- gsub('[+]', '', covid_table$Confirmed_Cases_New)
- covid_table$Recoveries_New <- gsub('[+]', '', covid_table$Recoveries_New)
- covid_table$Deaths_New <- gsub('[+]', '', covid_table$Deaths_New)
- covid_table$PCR_New <- gsub('[+]', '', covid_table$PCR_New)

#Removing % from three variables

- covid_table$TPR <- gsub('[%]', '', covid_table$TPR)
- covid_table$RR <- gsub('[%]', '', covid_table$RR)
- covid_table$CFR <- gsub('[%]', '', covid_table$CFR)

Data wrangling: Part III
Converting “chr” variables as integers 1

- covid_table$Confirmed_Cases_Total <- as.integer(covid_table$Confirmed_Cases_Total)
- covid_table$Confirmed_Cases_New <- as.integer(covid_table$Confirmed_Cases_New)
- covid_table$Confirmed_Cases_Active <- as.integer(covid_table$Confirmed_Cases_Active)
- covid_table$Recoveries_Total <- as.integer(covid_table$Recoveries_Total)
- covid_table$Recoveries_New <- as.integer(covid_table$Recoveries_New)

Data wrangling: Part IV
Converting “chr” variables as integers 2

- covid_table$Deaths_Total <- as.integer(covid_table$Deaths_Total)
- covid_table$Deaths_New <- as.integer(covid_table$Deaths_New)
- covid_table$PCR_Total <- as.integer(covid_table$PCR_Total)
- covid_table$PCR_New <- as.integer(covid_table$PCR_New)

Data wrangling: Part IV
#Converting “chr” variables as numbers

- covid_table$TPR <- as.numeric(covid_table$TPR)

- covid_table$RR <- as.numeric(covid_table$RR)

- covid_table$CFR <- as.numeric(covid_table$CFR)

How to change “date” variable?

- The date is shown as “23 Jan”, “24 Jan”, “25 Jan” etc.
- You need to use as.Date function
- What is the default Date values to use this function?
- Can you use different format to covert?
- **This is an assignment for you!**

Also see these posts to know more on web scrapping with “rvest” in R:

- https://kyleake.medium.com/wikipedia-data-scraping-with-r-rvest-in-action-3c419db9af2d

- https://www.engineeringbigdata.com/web-scraping-wikipedia-world-population-rvest-r/

- https://stackoverflow.com/questions/33360634/how-to-scrape-data-from-wikipedia-using-r

Collecting Web Data – APIs & Web Scrapping

https://research.library.gsu.edu/c.php?g=1050939&p=7628916

- Web scraping is an extremely popular amongst researchers and web developers. The best example may be Google Search.

- When you use Google to find information, you are (in highly over simplified terms) not actually searching the "live" internet, but rather a database of webpages that Google has mapped.

- If Google is allowed to do it, why can't you!?

Collecting Web Data – APIs & Web Scrapping

https://research.library.gsu.edu/c.php?g=1050939&p=7628916

1. Research Ethics

- Is the data you are collecting potentially sensitive information?
- If you are scraping user-comments from a social media website, are the users aware that their comments are visible to you or others?
- Are the users fully or partially aware of how their comments and data may be used?
- Do the users have an expectation of anonymity or confidentiality?
- Do the users represent marginalized or at-risk group?
- Does your research pose any form of potential risk to the users who supplied the data you are using?

Collecting Web Data – APIs & Web Scrapping

https://research.library.gsu.edu/c.php?g=1050939&p=7628916

2. Public vs. Protected Content

- Are the webpages you are collecting data from freely and publicly visible? OR...
- Are the webpages you are collecting data password-protected, requiring you to log into the website?
- If webpages and content are password-protected, does the website require you to adhere to a "Terms of Service", "Terms of Use" or other type of agreement in order to access and use the website?
  - Often these agreements explicitly forbid systematic web-scraping activities

Collecting Web Data – APIs & Web Scrapping

https://research.library.gsu.edu/c.php?g=1050939&p=7628916

3. Copyright & Commercial Activity

- Are you violating copyright as part of your overall as part of your web-scraping activities?

- Are you reproducing the data or contents of webpages on your own website or in another medium?

- If you are reproducing or embedding the content in some way, do you have the site owner's permission?

Collecting Web Data – APIs & Web Scrapping

https://research.library.gsu.edu/c.php?g=1050939&p=7628916

4. Sustainability

- Are you systematically collecting large volumes of webpages at a high rate from the target website?

- Are you systematically collecting on a repeating schedule at a rapid rate?

- Are you collecting webpages from the website in such a way that poses commercial and/or technical risk to the technical operations of the target website?

More on “ethical issues” with the use of web scrapping/ web APIs are here:

- https://towardsdatascience.com/ethics-in-web-scraping-b96b18136f01

- https://blogs.mulesoft.com/api-integration/strategy/ethics-of-apis/

- **Self-learning!**

Question/Queries?

Thank you!

@shitalbhandary