

Statistical Computing with R

Masters in Data Science 503 (S6)

Second Batch, SMS, TU, 2023

Shital Bhandary

Associate Professor

Statistics/Bio-statistics, Demography and Public Health Informatics

Patan Academy of Health Sciences, Lalitpur, Nepal

Faculty, Data Analysis and Decision Modeling, MBA, Pokhara University, Nepal

Faculty, FAIMER Fellowship in Health Professions Education, India/USA.

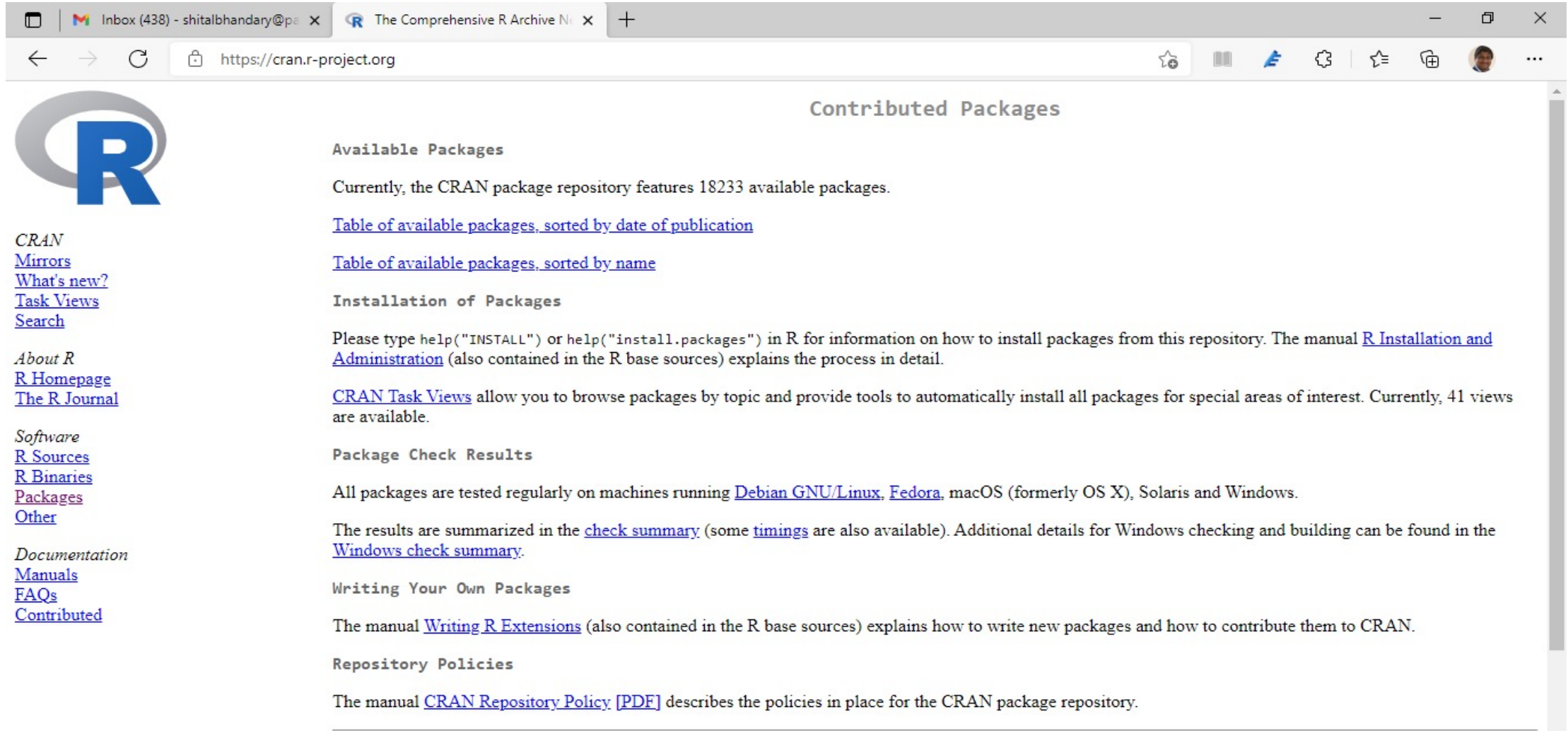
Review Preview (Unit 2, Part 1)

- 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 **Comprehensive R Archive Network**, 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.

Available “packages” from CRAN:



The screenshot shows a web browser window with the CRAN website. The browser's address bar displays 'https://cran.r-project.org'. The page features the CRAN logo on the left, a sidebar with navigation links, and a main content area with sections on contributed packages, available packages, installation, package check results, writing packages, and repository policies.

Contributed Packages

Available Packages

Currently, the CRAN package repository features 18233 available packages.

[Table of available packages, sorted by date of publication](#)

[Table of available packages, sorted by name](#)

Installation of Packages

Please type `help("INSTALL")` or `help("install.packages")` in R for information on how to install packages from this repository. The manual [R Installation and Administration](#) (also contained in the R base sources) explains the process in detail.

[CRAN Task Views](#) allow you to browse packages by topic and provide tools to automatically install all packages for special areas of interest. Currently, 41 views are available.

Package Check Results

All packages are tested regularly on machines running [Debian GNU/Linux](#), [Fedora](#), macOS (formerly OS X), Solaris and Windows.

The results are summarized in the [check summary](#) (some [timings](#) are also available). Additional details for Windows checking and building can be found in the [Windows check summary](#).

Writing Your Own Packages

The manual [Writing R Extensions](#) (also contained in the R base sources) explains how to write new packages and how to contribute them to CRAN.

Repository Policies

The manual [CRAN Repository Policy \[PDF\]](#) describes the policies in place for the CRAN package repository.

CRAN

- [Mirrors](#)
- [What's new?](#)
- [Task Views](#)
- [Search](#)

About R

- [R Homepage](#)
- [The R Journal](#)

Software

- [R Sources](#)
- [R Binaries](#)
- [Packages](#)
- [Other](#)

Documentation

- [Manuals](#)
- [FAQs](#)
- [Contributed](#)

“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](#)

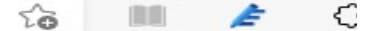
OS X El Capitan binaries of contributed packages

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!

not secure | www.bioconductor.org

[Home](#)[Install](#)[Help](#)[Developers](#)[About](#)

Search:

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.

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**

```
install.packages("devtools")
```

```
library("devtools")
```

```
devtools::install_github("klutometis/roxygen")
```

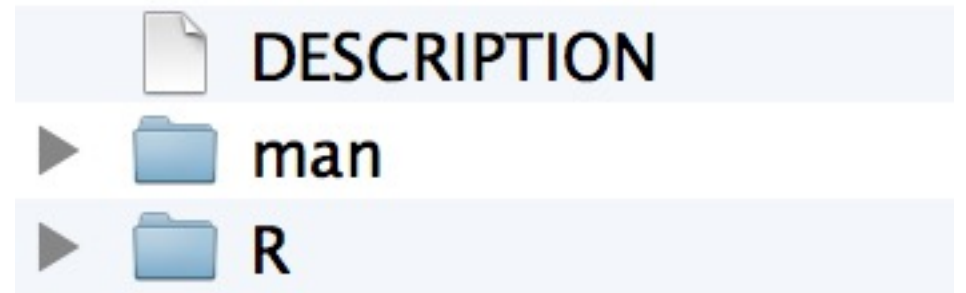
```
library(roxygen2)
```

How to develop a package in R?

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

- **Step 1: Creating your package directory**

```
setwd("parent_directory")  
create("cats")
```



- If you look in your parent directory, you will now have a folder called cats, and in it you will have two folders: man & R and, one file called DESCRIPTION.
- You should edit the DESCRIPTION file to include all of your contact information

How to develop a package in R?

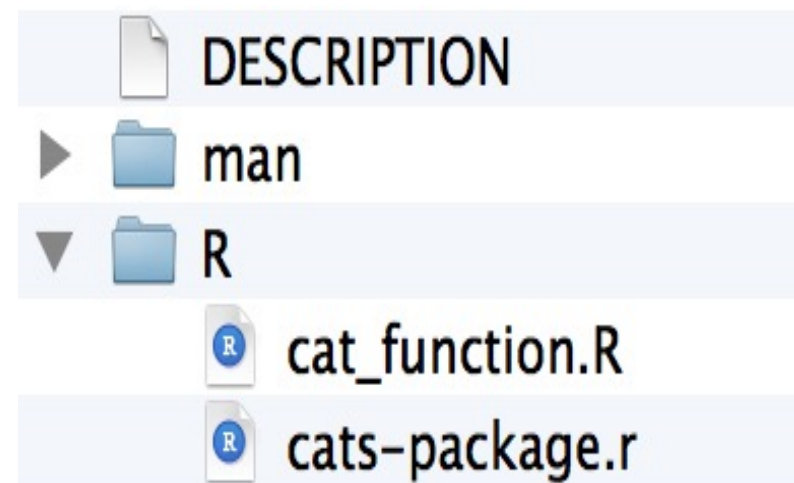
<https://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/>

- **Step 2: Add functions**

- For example:

```
cat_function <- function(love=TRUE){  
  if(love==TRUE){  
    print("I love cats!")  
  }  
  else {  
    print("I am not a cool person.")  
  }  
}
```

- Save this as a `cat_function.R` to your R directory.



How to develop a package in R?

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

- **Step 3: Add documentation**

```
1  #' A Cat Function
2  #'
3  #' This function allows you to express your love of cats.
4  #' @param love Do you love cats? Defaults to TRUE.
5  #' @keywords cats
6  #' @export
7  #' @examples
8  #' cat_function()
9
10 cat_function <- function(love=TRUE){
11     if(love==TRUE){
12         print("I love cats!")
13     }
14     else {
15         print("I am not a cool person.")
16     }
17 }
```

How to develop a package in R?

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

- **Step 4: Process your documentation**

- Now you need to create the documentation from your

- `setwd("./cats")`

- `document()`

- This automatically adds in the .Rd files to the man directory, and adds a NAMESPACE file to the main directory.

- More here: <https://r-pkgs.org/description.html>

This is what you will see:

		DESCRIPTION
▼		man
		cat_function.Rd
		cats.Rd
		NAMESPACE
▼		R
		cat_function.R
		cats-package.r

How to develop a package in R?

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

- Step 5: Install
- Now it is as simple as installing the package! You need to run this from the parent working directory that contains the **cats** folder.
- `setwd("../")`
- `install("cats")`
- Now you have a real, live, functioning R package.

For example, try typing ?cat_function

```
cat_function {cats}
```

A Cat Function

Description

This function allows you to express your love of cats.

Usage

```
cat_function(love = TRUE)
```

Arguments

love Do you love cats? Defaults to TRUE.

Examples

```
cat_function()
```

[Package *cats* version 0.1 [Index](#)]

Step 6: Make the package a GitHub repo (Bonus!)

- The benefit, however, to putting your package onto GitHub is that you can use the `devtools::install_github()` function to install your new package directly from the GitHub page.
- `install_github('cats','github_username')`

Step 7.1: Infinity

- This is where the benefit of having the package pulled together really helps.
- You can flesh out the documentation as you use and share the package.
- You can add new functions the moment you write them, rather than waiting to see if you'll reuse them.
- You can divide up the functions into new packages.
- The possibilities are endless!

Note: "... best products are built in small steps, not by waiting for a perfect final product to be created"

Step 7.2: Iterate

The best products are built in small steps, not by waiting for a perfect final product to be created

This concept is called the minimum viable product — it's best to get a project started and improve it through iteration.

R packages can seem like a big, intimidating feat, and they really shouldn't be. **The minimum viable R package is a package with just one function!**

Reading 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
- JSON files: rjson, 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","5/2  
1/2013","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
- `Install.packages("rjson")`

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

- `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 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 more can you do with the food_market data?

- Try: `table(Names)`
- Try: `table(V19)`
- 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!

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()`

How to do complex web scrapping?


Inspect the HTML elements!

The screenshot shows a web browser window with the Wikipedia article "COVID-19 pandemic in Nepal" open. The browser's address bar shows the URL `en.wikipedia.org/wiki/COVID-19_pandemic_in_Nepal`. The page features the Wikipedia logo and a sidebar with navigation links. A right-click context menu is open over the article text, with the "Inspect" option highlighted. To the right of the article, a summary box titled "COVID-19 pandemic in Nepal" provides key statistics.

COVID-19 pandemic in Nepal

Disease	COVID-19
Virus strain	SARS-CoV-2
Location	Nepal
First outbreak	Wuhan, Hubei, China
Index case	Kathmandu, Bagmati Province
Arrival date	9 January 2020 (1 year, 7 months, 4 weeks and 1 day)
Date	23 January 2020
Confirmed cases	783,910 (18 September) ^[1]
Active cases	▲ 25,082 (18 September) ^[1]
Recovered	747,800 (18 September) ^[1]
Deaths	11,028 (18 September) ^[1]

What to do now?



WIKIPEDIA
The Free Encyclopedia

- Main page
- Contents
- Current events
- Random article
- About Wikipedia
- Contact us
- Donate

Contribute

- Help
- Learn to edit
- Community portal
- Recent changes
- Upload file

Tools

- What links here
- Related changes
- Special pages
- Permanent link
- Page information
- Cite this page


Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Search Wikipedia

COVID-19 pandemic in Nepal

From Wikipedia, the free encyclopedia



This article needs to be **updated**. The reason given is: Almost everything except the stat in the infobox is outdated.. Please help update this to reflect recent events or newly available information. *(September 2021)*

The **COVID-19 pandemic in Nepal** is part of the worldwide pandemic of coronavirus disease 2019 (COVID-19) caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The first case in Nepal was confirmed on 23 January 2020 when a 31-year-old student, who had returned to Kathmandu from Wuhan on 9 January, tested positive for the disease.^[2] It was also the first recorded case of COVID-19 in South Asia.^[3] Nepal's first case of local transmission was confirmed on 4 April in Kailali District. The first death occurred on 14 May. A country-wide lockdown came into effect on 24 March 2020, and ended on 21 July 2020.^[4] As of 18 September 2021, the Ministry of Health and Population

COVID-19 pandemic in Nepal	
Disease	COVID-19
Virus strain	SARS-CoV-2
Location	Nepal
First outbreak	Wuhan, Hubei, China
Index case	Kathmandu, Bagmati Province
Arrival date	9 January 2020 (1 year, 7 months, 4 weeks and 1 day)
Date	23 January 2020
Confirmed cases	783,910 (18 September) ^[1]
Active cases	▲ 25,082 (18 September) ^[1]
Recovered	747,800 (18 September) ^[1]

1 The tidy text format | Text Mining with R
https://www.tidytextmining.com/tidytext.html

```
<html class="client-js ve-available" lang="en" dir="ltr">
<head>...</head>
<body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject mw-editable page-COVID-19_pandemic_in_Nepal rootpage-COVID-19_pandemic_in_Nepal skin-vector action-view skin-vector-legacy">
  <div id="mw-page-base" class="noprint"></div>
  <div id="mw-head-base" class="noprint"></div>
  <div id="content" class="mw-body" role="main">
    <a id="top"></a>
    <div id="siteNotice">...</div>
    <div class="mw-indicators">...</div>
    ... <h1 id="firstHeading" class="firstHeading">...</h1> == $0
    <div id="bodyContent" class="vector-body">...</div>
  </div>
... html.client-js.ve-available body.mediawiki.ltr.sitedir-ltr.mw-hide-empty-elt.ns-0.ns-subject:r ...
```

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

element.style {

.mw-body .firstHeading {
 overflow: visible;
}

.mw-body h1, .mw-body-content h1 {
 font-size: 1.8em;
}

.mw-body h1, .mw-body-content h1, .mw-body-content h2 {
 margin-bottom: 0.25em;
 padding: 0;
 font-family: 'Linux Libertine', 'Georgia', 'Times', serif;
 line-height: 1.3;
}

h1 {
 font-size: 180%;


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

Inbox (58) - shitalbhandary@gmail.com x COVID-19 pandemic in Nepal - V x +

en.wikipedia.org/wiki/COVID-19_pandemic_in_Nepal

Apps YouTube Maps Gmail How To Remove A... cov-glue.cvr.gla.ac... News Translate 1 The tidy text form... Welcome to Text Mi... Reading list

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:

- `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**
- Find the node number with class="wikitable sortable" as per inspect
 - `[7] <table class="wikitable sortable" style="text-align:right">\n<caption>\n</caption>\n<tbody ...`

Let's grab this node in R as "covid_table"

- `covid_table <- wiki_page %>% html_nodes("table") %>% .[7] %>%
html_table() %>% .[[1]]`
 - `str(covid_table)` **#tibble [496 x 14]; tibble = fast data frame!**
- #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)`

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”

Check if this works or not!

You can use other method too!

- `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?

```
15 colnames(covid_table)
16 names(covid_table)[names(covid_table) == "Date_Date"] = "Date"
17 names(covid_table)[names(covid_table) == "Confirmed cases_Total"] = "Confirmed_Cases_T"
18 names(covid_table)[names(covid_table) == "Confirmed cases_New"] = "Confirmed_Cases_New"
19 names(covid_table)[names(covid_table) == "Confirmed cases_Active"] = "Confirmed_Cases_"
20 names(covid_table)[names(covid_table) == "RT-PCR tests_Total"] = "PCR_Total"
21 names(covid_table)[names(covid_table) == "RT-PCR tests_New"] = "PCR_New"
22 names(covid_table)[names(covid_table) == "TPR_TPR"] = "TPR"
23 names(covid_table)[names(covid_table) == "RR_RR"] = "RR"
24 names(covid_table)[names(covid_table) == "CFR_CFR"] = "CFR"
25 names(covid_table)[names(covid_table) == "Ref._Ref."] = "Ref"
26 colnames(covid_table)
```

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)`

#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)

#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)`

#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!**

#OR, create “date2” variable as follows:

- `date2 = seq(as.Date('2020-1-23'), by='days', length.out = 495)`
- `covid_table = cbind(covid_table, date2)`
- `str(covid_table)`
- `summary(covid_table)`
- **Save this cleaned data as “covid_table.csv” file for next class!**

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>

- **Before assuming that your project is both legal and ethical, you need to ask yourself a lot of questions regarding the nature, scope, and purpose of your web scraping activities.**
- **If your web scraping project is focused on collecting data that is in the public-domain from publicly available websites and individual datum do not connect directly to individual humans in the real world and your web scraping activities represent minimal technical burden on the part of the website owner, then you are in the clear!**

Collecting Web Data – APIs & Web Scrapping

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

- However, in truth, most web scraping research projects are not so clearly defined.
- While you will have to make the determination yourself regarding whether your project is both legal and ethical, the questions in the subsequent slides are meant to prompt the kind of thinking that may not be immediately obvious the first time you start scraping the internet for data.

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