Statistical Computing with R
Masters in Data Science 503 (S1)
Second Batch, SMS, TU, 2023

Shital Bhandary

Associate Professor

Community/Public Health (Statistics, Demography and Medical Education)

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

Course Description:

- This is an outcome based course to introduce basic programming in R software followed by use of R software for Statistical Computing.
- It focuses on the use of R software for data manipulation, data summary/data visualization, models (supervised and unsupervised learnings) and communicate the findings

Learning outcomes:

- Understand, use and apply R software for basic programming (program)
- Understand, use and apply R software for data manipulation (wrangle)
- Understand, use and apply R software for data summary and visualization (explore)
- Understand, use and apply R software for supervised learning (model)
- Understand, use and apply R software for unsupervised learning (model)
- Understand, use and apply R software to communicate findings (communicate).

Course delivery and assessment in/for/of learning (Classroom and MS Teams):

- Didactic session
- Individual/Group work
- Project works
- Practical/Lab session
- Presentation
  - Individual
  - Group
- Assignments
- Report

Course books:

- Required (core):
  - Wichham Hadley & Gloremund Garrette (2017). R for Data Science. O'Reilly Media Inc: Sebastopol, Canada. Available for free in HTML from this website: https://r4ds.had.co.nz/index.html

- Suggested (non-core):
  - Mailund Thomas (2017). Beginning Data Sciences in R: Data Analysis, Visualization, and Modelling for the Data Scientists. Apress: Aarhus, Denmark.

  - Goh Eric & Hui Ming (2019). Learn R for Applied Statistics. Apress: Singapore

Tools for this course:

- Software
  - **R**
  - Latest version recommended

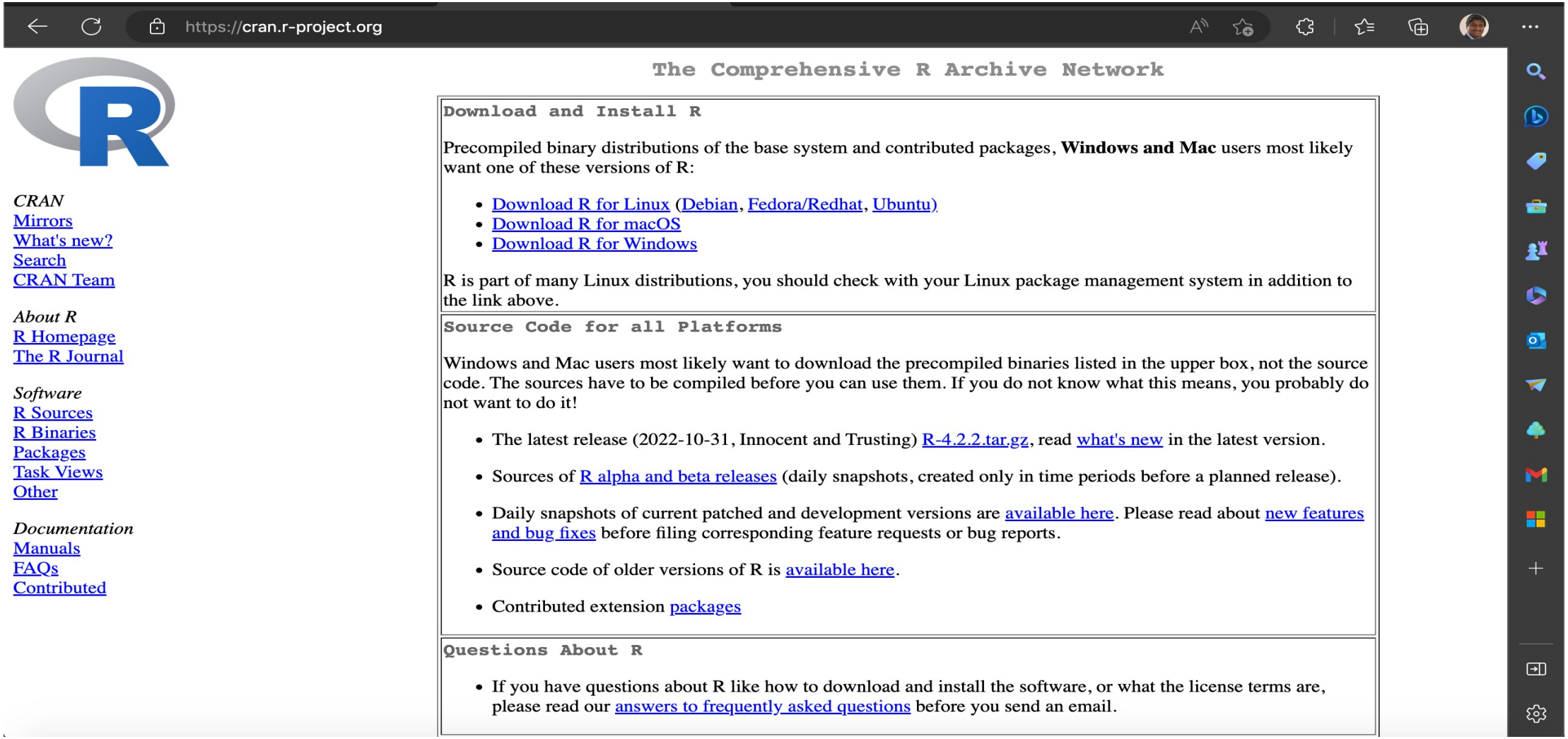
- IDE
  - **R Studio**
  - Latest version recommended

- Packages
  - **Base**
  - ***Recommended (tidyverse packages for data sciences)***
  - As per the requirement of the course

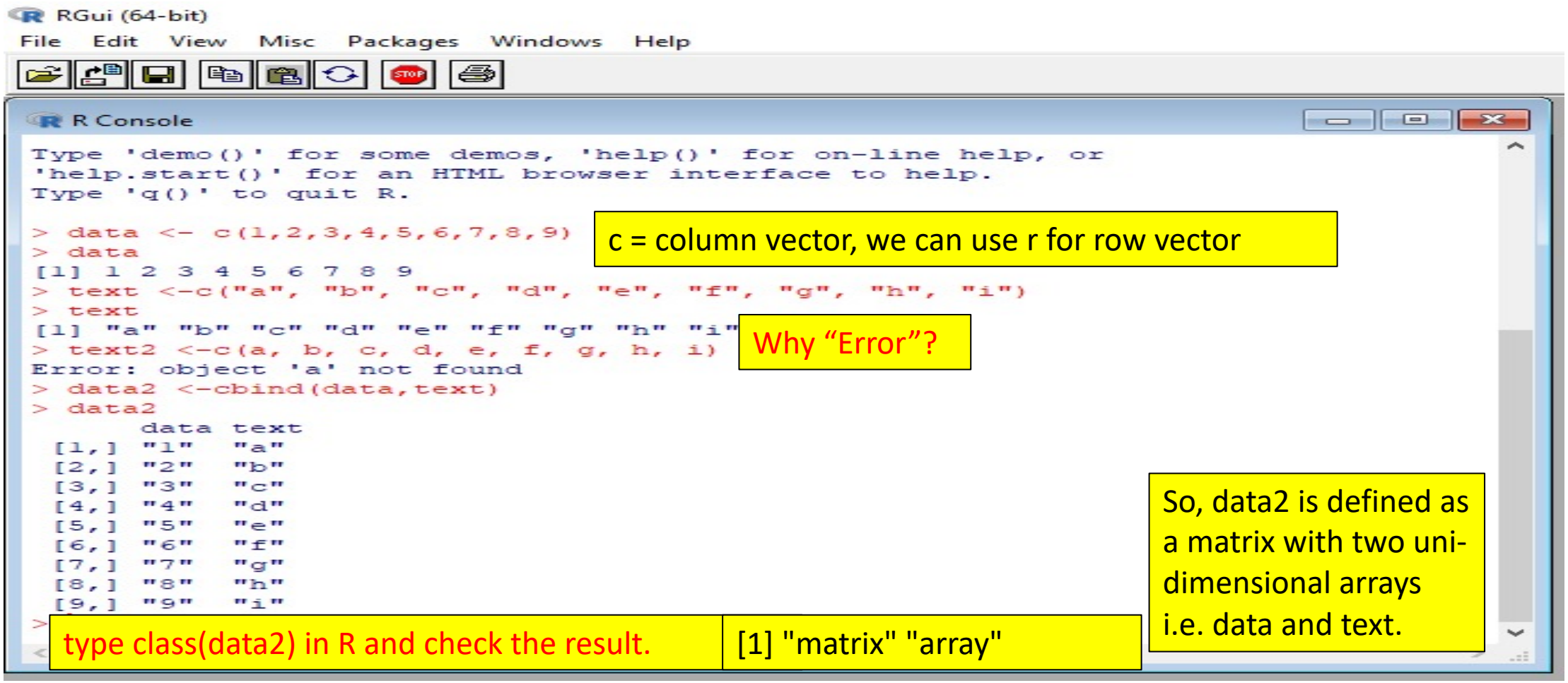
R: History (https://mran.microsoft.com/documents/what-is-r)

- First implemented in 1990's by Ross Ihaka and Robert Gentleman at University of Auckland, New Zealand
- Established as an open source project in 1995 by Ross Ihaka
- R project is managed by R core group since 1997
- R 1.0.0 was released in February 2000 (I saw it on September 2000)
- R is closely modeled on the S language for statistical computing conceived by John Chambers, Rick Baker, Trevor Hastie, Allan Wilks and others at Bell Labs in mid 1970s, and made publicly available in 1980s. (But, S and Splus software are commercial)
- Read "R: Past and Future History by Ross Ihaka" for more

R installation: CRAN Website

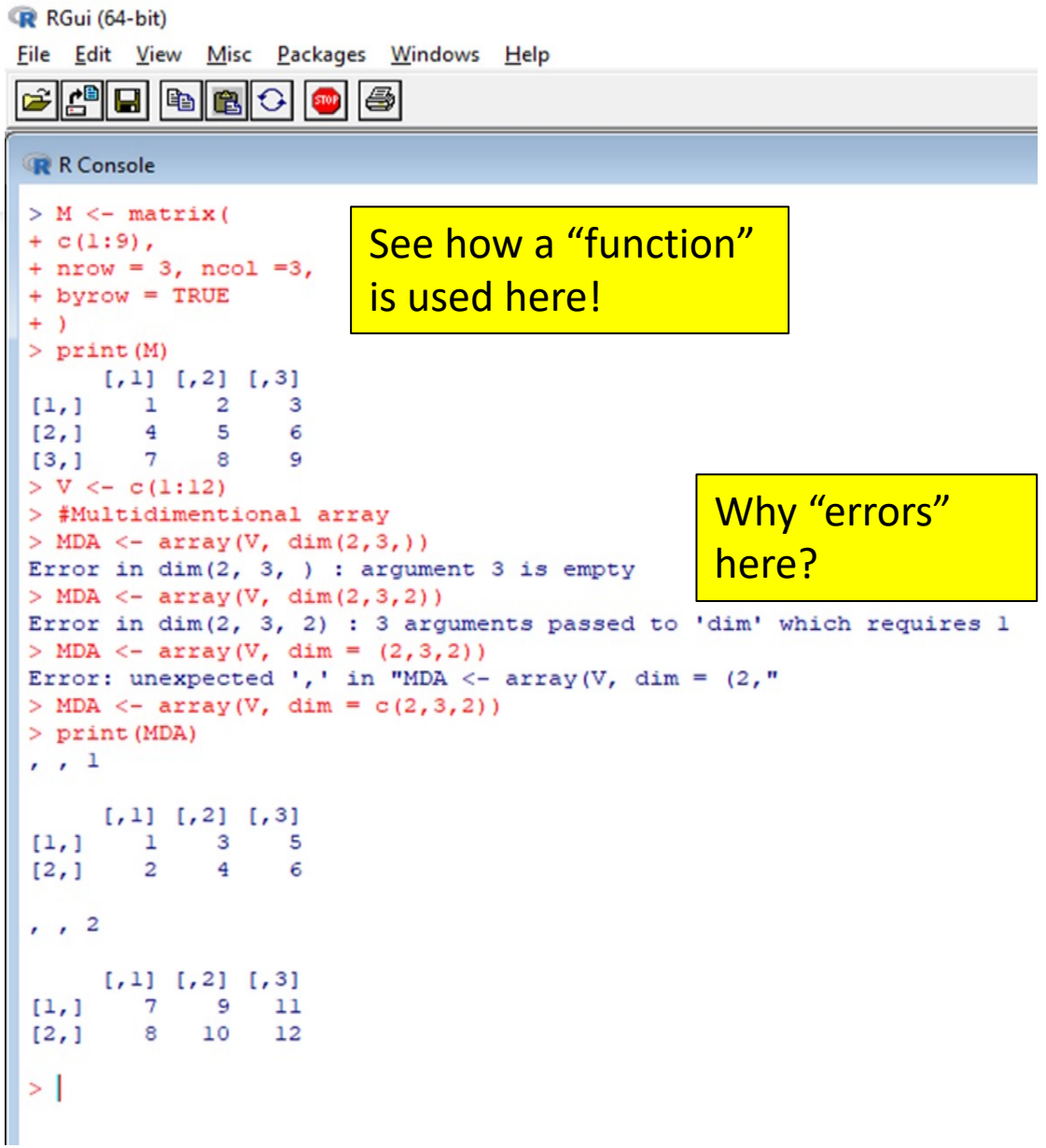


(Simple) Data entry in R:



Arrays and Matrices in R:

Arrays	Matrices
Arrays can contain greater than or equal to 1 dimensions.	Matrices contains 2 dimensions in a table like structure.
Array is a homogeneous data structure.	Matrix is also a homogeneous data structure.
It is a singular vector arranged into the specified dimensions.	It comprises of multiple equal length vectors stacked together in a table.
array() function can be used to create matrix by specifying the third dimension to be 1.	**matrix()** function however can be used to create at most 2-dimensional array.
Arrays are superset of matrices.	Matrices are a subset, special case of array where dimensions is two.
Limited set of collection-based operations.	Wide range of collection operations possible.
Mostly, intended for storage of data.	Mostly, matrices are intended for data transformation.



https://www.geeksforgeeks.org/array-vs-matrix-in-r-programming/

Quick Think!

Individual assignment at MS Teams!

- What is “list” in R?
- How to create a list containing strings, numbers, vectors and a logical values in R?
- How to name the list elements in R?
- How to assess list elements in R?
- How to manipulate list elements in R?
- How to convert list to vectors in R?

We use “Data frame” a lot in R: (https://www.educba.com/data-frames-in-r/)

- Data frames in R language are the type of data structure that is used to store data in a tabular form which is of two dimensional.
- The data frames are special categories of list data structure in which the components are of equal length.
- R languages support the built-in function i.e. data.frame() to create the data frames and assign the data elements.

We use “Data frame” a lot in R because: (https://www.educba.com/data-frames-in-r/)

- R language supports the data frame name to modify and retrieve data elements from the data frames.
- Data frames in R structured as column name by the component name also, structured as rows by the component values.
- Data frames in R is a widely used data structure while developing the machine learning models in data science projects.

Creating a simple “data.frame” in R:

- df <- data.frame(x=c(1,2,3),y=c(2,3,4),z=c(3,4,5))
- df

```
  x y z
1 1 2 3
2 2 3 4
3 3 4 5
```

- class(df)

```
[1] "data.frame"
```

A small but realistic data frame and its use: (https://www.tutorialspoint.com/r/r_data_frames.htm)

#create data frame

- emp.data <- data.frame(
  emp_id = c(1:5),
  emp_name =c("Rick", "Dan", "Michelle", "Ryan", "Gary"),
  salary = c(623.3, 515.2, 611.0, 729.0, 845.25),
  start_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11", "2015-03-27")),
  stringAsFactors = FALSE

)

#Print the data

- print(emp.data)

Sample Data frame in R:

Structure and Summary of Sample Data Frame in R:

```
> # Get the structure of the data frame.
> str(emp.data)
'data.frame':   5 obs. of  4 variables:
 $ emp_id    : int  1 2 3 4 5
 $ emp_name  : chr  "Rick" "Dan" "Michelle" "Ryan" ...
 $ salary    : num  623 515 611 729 843
 $ start_date: Date, format: "2012-01-01" "2013-09-23" "2014-11-15" ...
> # Print the summary.
> print(summary(emp.data))
     emp_id    emp_name             salary        start_date
 Min.   :1   Length:5           Min.   :515.2   Min.   :2012-01-01
 1st Qu.:2   Class :character   1st Qu.:611.0   1st Qu.:2013-09-23
 Median :3   Mode  :character   Median :623.3   Median :2014-05-11
 Mean   :3                      Mean   :664.4   Mean   :2014-01-14
 3rd Qu.:4                      3rd Qu.:729.0   3rd Qu.:2014-11-15
 Max.   :5                      Max.   :843.2   Max.   :2015-03-27
> |
```

Extract part of data from Data Frame in R: (Very useful when working with large data)

```
> # Extract Specific columns.
> result <- data.frame(emp.data$emp_name,emp.data$salary)
> print(result)
  emp.data.emp_name emp.data.salary
1              Rick           623.30
2               Dan           515.20
3          Michelle           611.00
4              Ryan           729.00
5              Gary           843.25
> # Extract first two rows.
> result <- emp.data[1:2,]
> print(result)
  emp_id emp_name salary start_date
1      1      Rick  623.3 2012-01-01
2      2       Dan  515.2 2013-09-23
> # Extract 3rd and 5th row with 2nd and 4th column.
> result <- emp.data[c(3,5),c(2,4)]
> print(result)
  emp_name start_date
3 Michelle 2014-11-15
5     Gary 2015-03-27
> 
```

Add a new column in existing Data Frame:

```
> # Add the "dept" coulmn.
> emp.data$dept <- c("IT","Operations","IT","HR","Finance")
> v <- emp.data
> print(v)
  emp_id emp_name salary start_date       dept
1      1     Rick 623.30 2012-01-01         IT
2      2      Dan 515.20 2013-09-23 Operations
3      3 Michelle 611.00 2014-11-15         IT
4      4     Ryan 729.00 2014-05-11         HR
5      5     Gary 843.25 2015-03-27    Finance
```

Expand data frame in R (Adding cases):

```
> # Create the first data frame.
> emp.data <- data.frame(
+    emp_id = c (1:5),
+    emp_name = c("Rick","Dan","Michelle","Ryan","Gary"),
+    salary = c(623.3,515.2,611.0,729.0,843.25),
+
+    start_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",
+       "2015-03-27")),
+    dept = c("IT","Operations","IT","HR","Finance"),
+    stringsAsFactors = FALSE
+ )
> # Create the second data frame
> emp.newdata <- data.frame(
+    emp_id = c (6:8),
+    emp_name = c("Rasmi","Pranab","Tusar"),
+    salary = c(578.0,722.5,632.8),
+    start_date = as.Date(c("2013-05-21","2013-07-30","2014-06-17")),
+    dept = c("IT","Operations","Fianance"),
+    stringsAsFactors = FALSE
+ )
```

Already defined!
Cases: 1 to 5

Added data!
Cases: 6 to 8

Expand data frame in R (rbind is used):

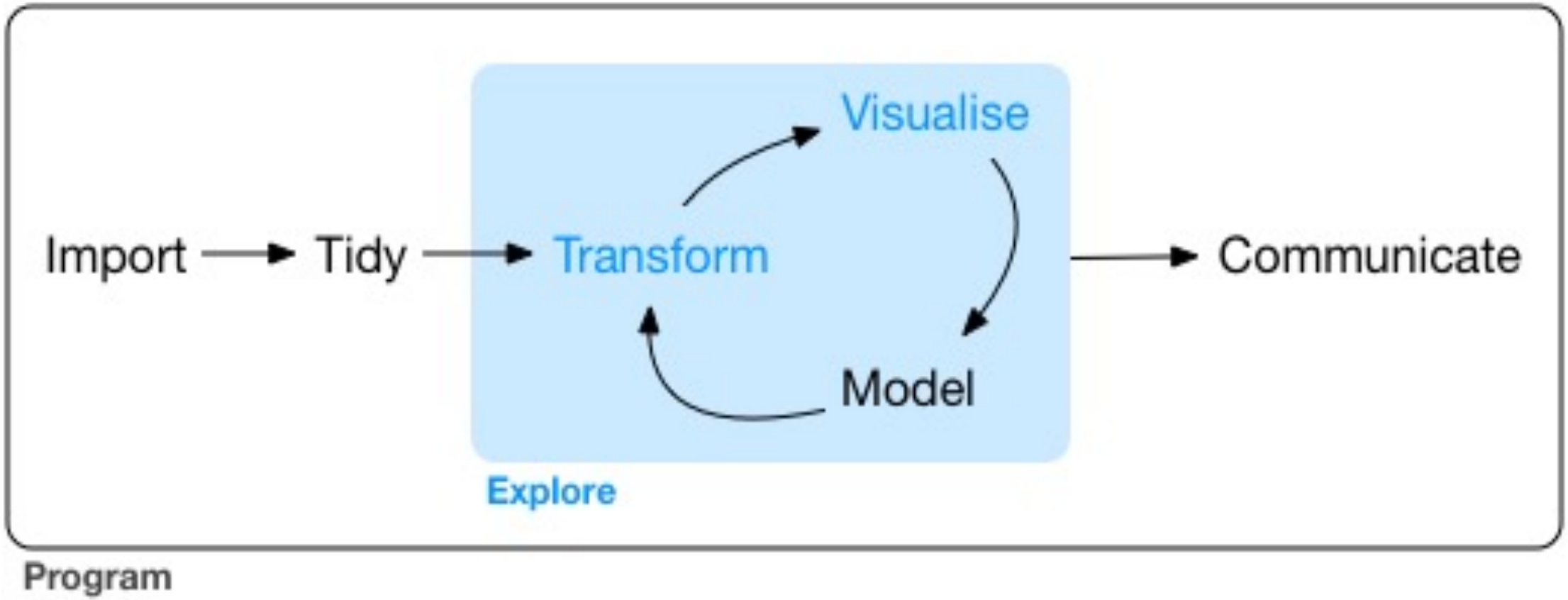
```
> # Bind the two data frames.
> emp.finaldata <- rbind(emp.data,emp.newdata)
> print(emp.finaldata)
  emp_id emp_name salary start_date          dept
1      1     Rick 623.30 2012-01-01            IT
2      2      Dan 515.20 2013-09-23 Operations
3      3 Michelle 611.00 2014-11-15            IT
4      4     Ryan 729.00 2014-05-11            HR
5      5     Gary 843.25 2015-03-27       Finance
6      6    Rasmi 578.00 2013-05-21            IT
7      7   Pranab 722.50 2013-07-30 Operations
8      8    Tusar 632.80 2014-06-17     Fianance
> |
```

Questions/queries?

- You can use this website to find more resources for R:

- https://stackoverflow.com/questions/tagged/r

Introduction to data science with R:



Import: First step of Data Science!

- What?
  - Data (What is?)
  - Big data (What is?)

- How?
  - Code (How to write?)
  - Packages (How to install, load and use?)
  - Any package or specific package for data science?

Import data in R: Text files

- Base
  - read.table(), read.delim(), read.csv(), read.csv2()
  - table = Reads text file e.g. data with 4 rows and 3 columns in R
  - delim = Tab separated values text file
  - csv = Comma separated values text file
  - csv2 = Semi-colon separate values text file

- Examples:
  - read.csv("C:/Documents/Users/Dell/Documents/covnep_252days.csv")
  - read.csv("covnep_252days.csv") works if this data is in the working directory **#getwd()**
  - read.csv(file.choose()) TO OPEN A NEW WINDOW TO READ FILE

Base read.csv function to read csv file:

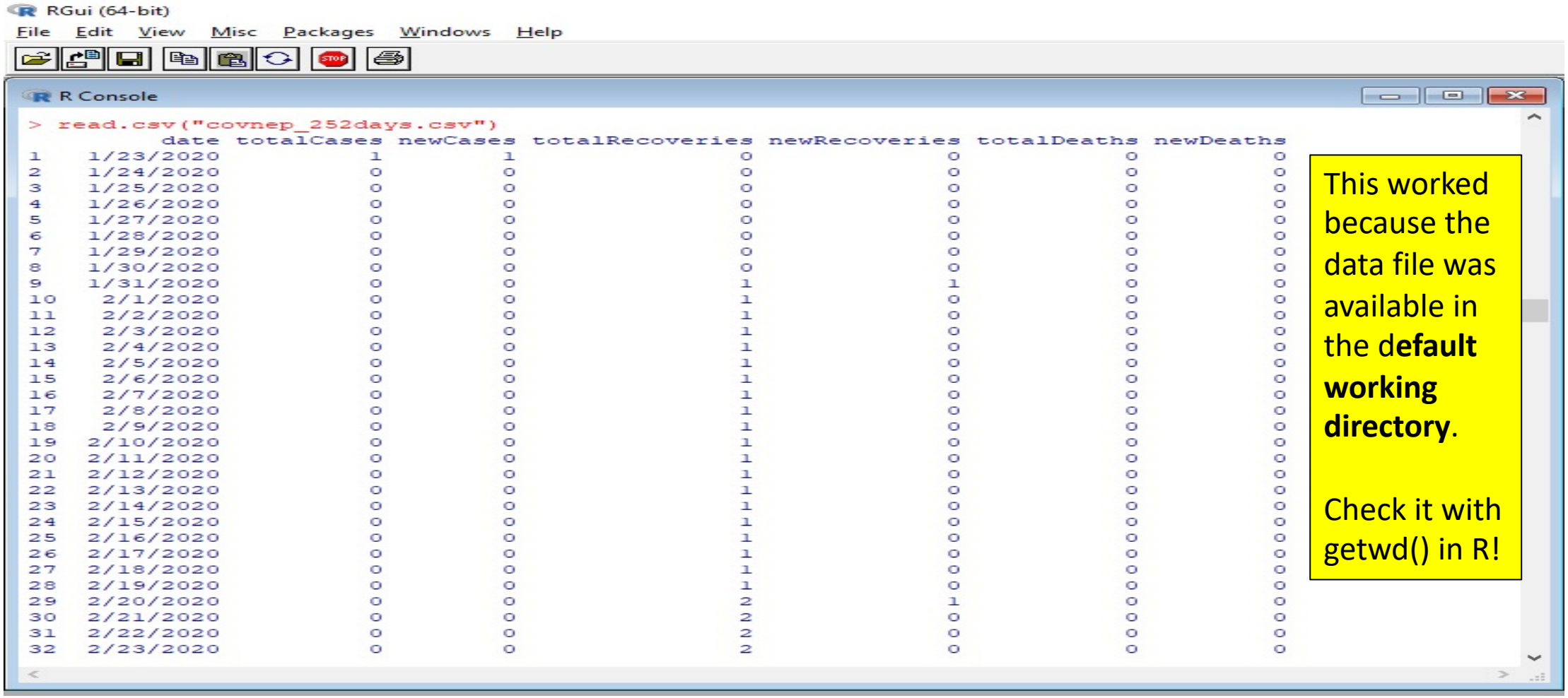
RGui (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
> read.csv("C:/Documents/covnep_252days.csv")
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file 'C:/Documents/covnep_252days.csv': No such file or directory
> read.csv("C:/Users/Dell/Documents/covnep_252days.csv")
         date totalCases newCases totalRecoveries newRecoveries totalDeaths newDeaths
1    1/23/2020          1        1               0             0           0         0
2    1/24/2020          0        0               0             0           0         0
3    1/25/2020          0        0               0             0           0         0
4    1/26/2020          0        0               0             0           0         0
5    1/27/2020          0        0               0             0           0         0
6    1/28/2020          0        0               0             0           0         0
7    1/29/2020          0        0               0             0           0         0
8    1/30/2020          0        0               0             0           0         0
9    1/31/2020          0        0               1             1           0         0
10    2/1/2020          0        0               1             0           0         0
11    2/2/2020          0        0               1             0           0         0
12    2/3/2020          0        0               1             0           0         0
13    2/4/2020          0        0               1             0           0         0
14    2/5/2020          0        0               1             0           0         0
15    2/6/2020          0        0               1             0           0         0
16    2/7/2020          0        0               1             0           0         0
17    2/8/2020          0        0               1             0           0         0
18    2/9/2020          0        0               1             0           0         0
19   2/10/2020          0        0               1             0           0         0
20   2/11/2020          0        0               1             0           0         0
21   2/12/2020          0        0               1             0           0         0
22   2/13/2020          0        0               1             0           0         0
23   2/14/2020          0        0               1             0           0         0
24   2/15/2020          0        0               1             0           0         0
25   2/16/2020          0        0               1             0           0         0
26   2/17/2020          0        0               1             0           0         0
27   2/18/2020          0        0               1             0           0         0
```

Base read.csv function to read csv file: Get summary of this “data.frame” in R!



This worked because the data file was available in the d**efault working directory**.

Check it with getwd() in R!

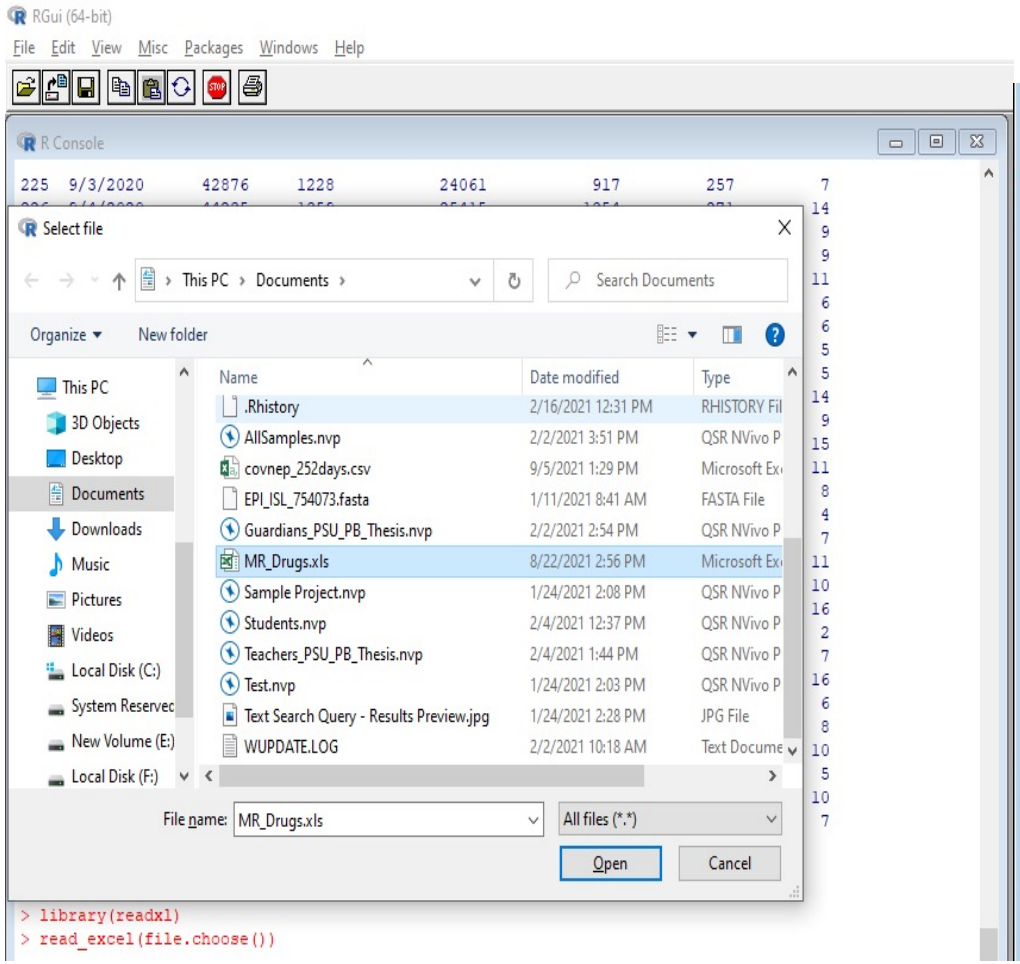
Import data in R: Excel files

- Packages:
  - “readxl”, “xlxs” packages

- How to use “readxl” package to read xls and xlxs excel files
  - Install.packages(“readxl”)
  - load(readxl)
  - my_data1 <- read_excel(“my_file.xls”)
  - my_data2 <- read_excel(“my_file.xlsx”)

- Use “xlxs” package if “readxl” package can’t read excel file with xlsx extensions!

Readxl package to read excel files:
Get summary of this “data.frame” in R!

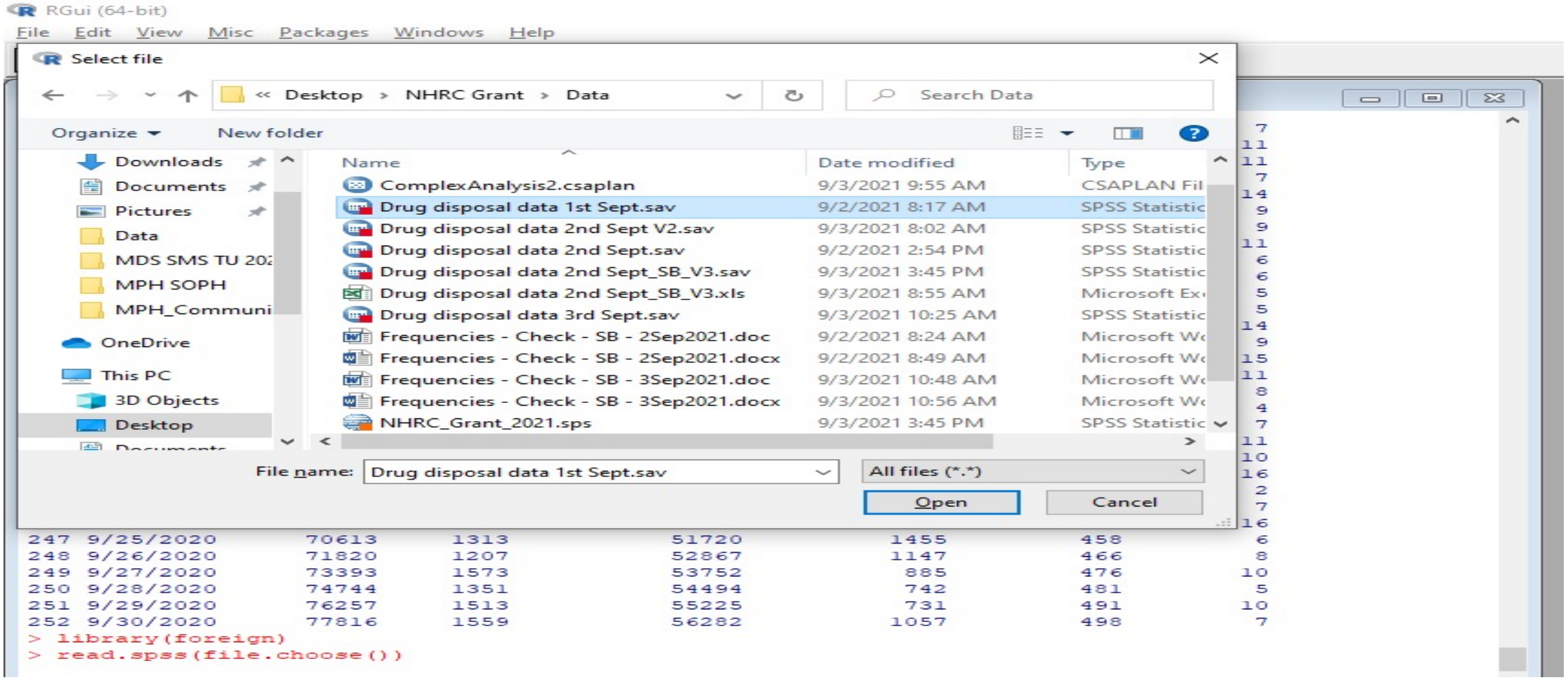



```
> library(readxl)
> read_excel(file.choose())
# A tibble: 972 x 27
      id   sex  city incol inco2 inco3 inco4 inco5 inco6 inco7 pinco1 pinco2 pinco3 pinco4 pinco5
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
 1  1001     2     1     0     0     0     0     0     1     0      6     -1     -1     -1     -1
 2  1002     2     1     0     1     0     0     0     0     0      2     -1     -1     -1     -1
 3  1003     2     1     0     0     0     0     0     1     0      6     -1     -1     -1     -1
 4  1004     2     1     0     1     0     0     0     0     0      2     -1     -1     -1     -1
 5  1005     2     1     0     0     0     0     0     0     1      7     -1     -1     -1     -1
 6  1006     2     1     1     1     0     0     0     0     0      2      1     -1     -1     -1
 7  1007     2     1     0     1     0     0     0     0     0      2     -1     -1     -1     -1
 8  1008     2     1     0     1     0     0     0     0     0      2     -1     -1     -1     -1
 9  1009     2     1     1     1     0     0     0     0     0      2      1     -1     -1     -1
10  1010     2     1     0     1     0     0     0     0     0      2     -1     -1     -1     -1
# ... with 962 more rows, and 12 more variables: pinco6 <dbl>, sinco1 <chr>, sinco2 <chr>,
#   sinco3 <chr>, sinco4 <chr>, sinco5 <chr>, sinco6 <chr>, crime1 <dbl>, crime2 <dbl>,
#   crime3 <dbl>, crime4 <dbl>, crime5 <dbl>
>
```

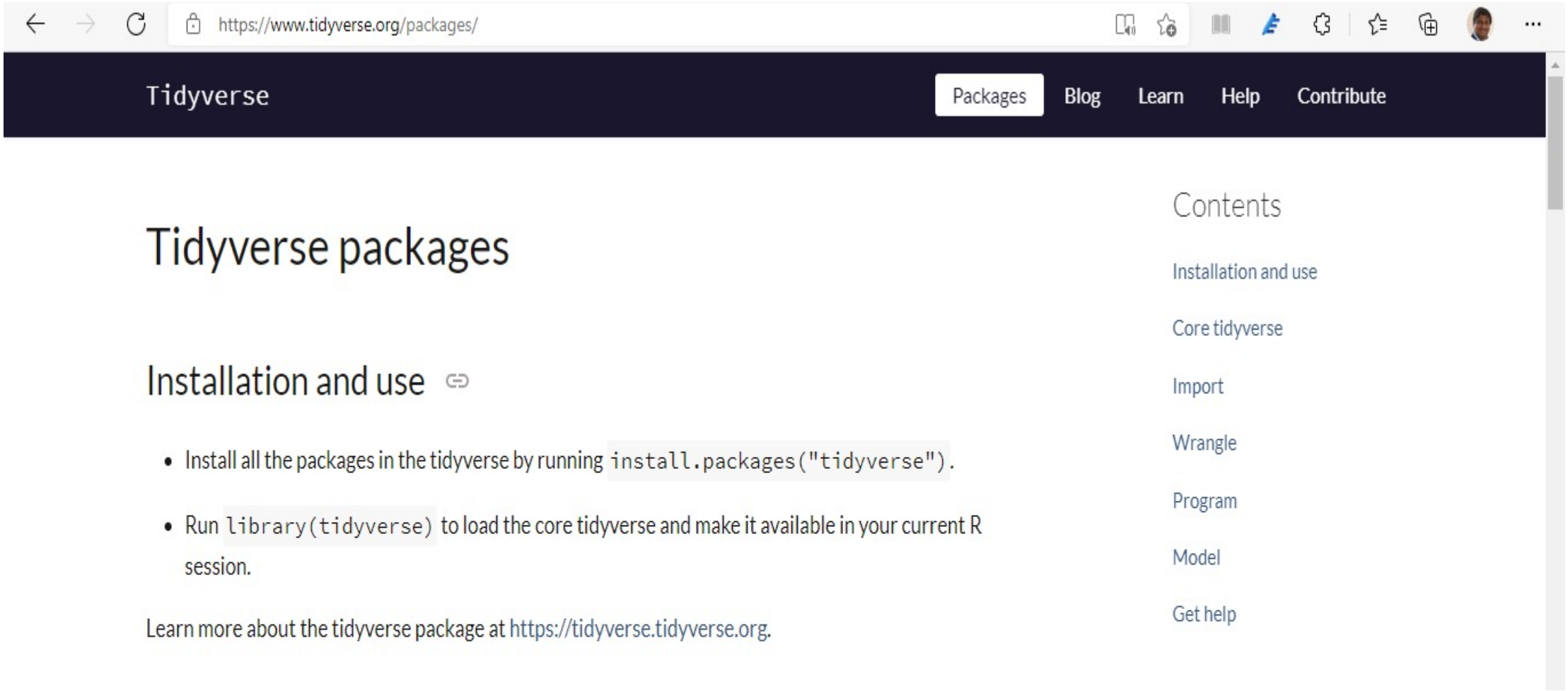
Import data in R: SPSS, Stata, Minitab files

- Packages:
  - “foreign”
- How to use “foreign” package?
  - Install the package in R using: install.packages(“foreign”) command
  - Load the package in R using: library(foreign) function
  - # Reads SPSS file with: read.spss(“datafile”, to.data.frame=TRUE)
  - # Reads Stata file with: read.stata(“datafile”, to.data.frame=T)
  - # Reads Minitab transport file with: read.mtp(“datafile”, to.data.frame=T)
- The “datafile” = “datafile.sav” or “datafile.dta” or “datafile.mtp” files must be in the working directory (Check with “getwd()” in R!)

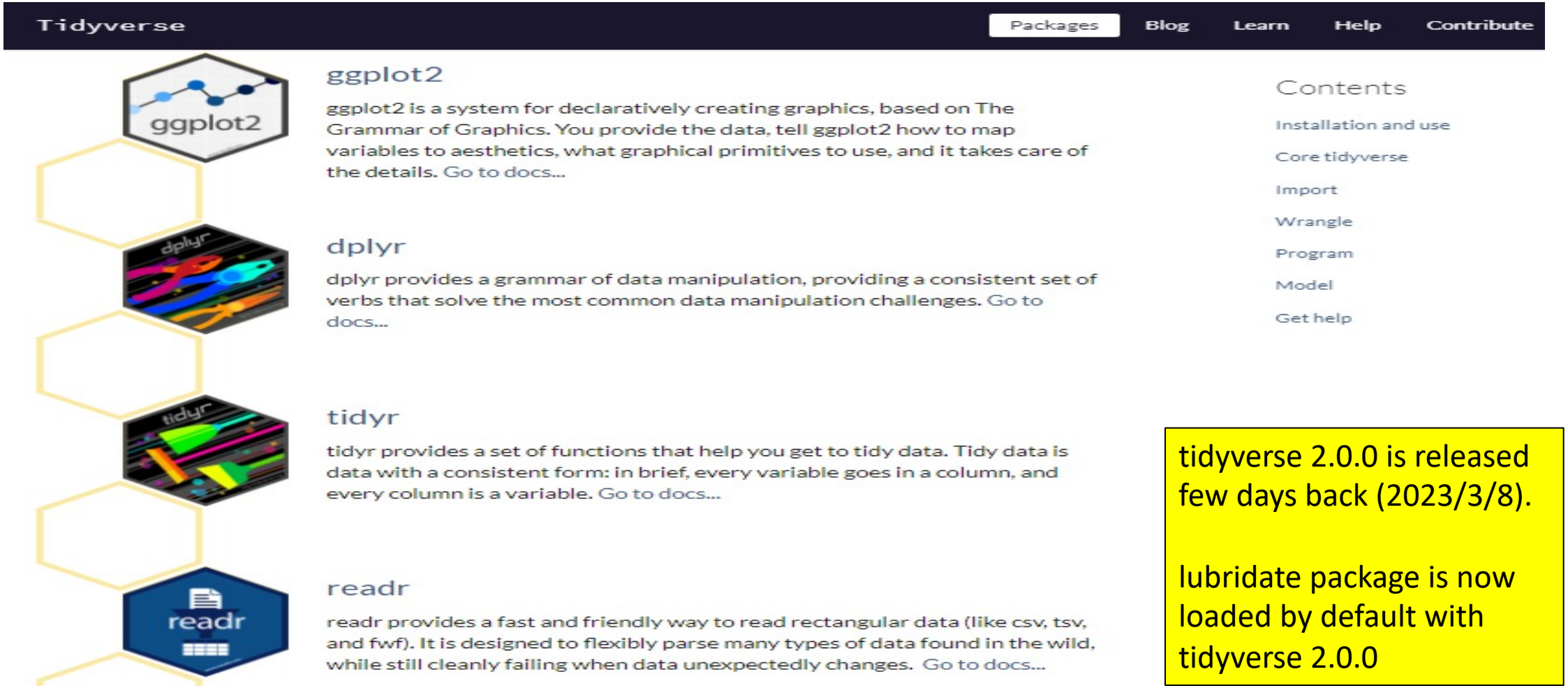
Reading SPSS with “Foreign” package:



The “tidyverse” package for data science:



Core “tidyverse” packages



tidyverse 2.0.0 is released few days back (2023/3/8).

lubridate package is now loaded by default with tidyverse 2.0.0

Import data in R: Self-Practice!

- Packages:
  - “readr”

- How to use “readr” package as stand-alone package?
  - Install the package in R using: install.packages(“readr”) command
  - Load the package in R using: library(readr) function
  - # Read tab separated values read_tsv(file.choose())
  - # Read comma (",") separated values read_csv(file.choose())
  - # Read semicolon (";") separated values read_csv2(file.choose())

Import data in R: SPSS, Stata and SAS files (Self-Practice)

- Packages:
  - “haven”
- How to use “haven” package?
  - It is part of the “tidyverse” package

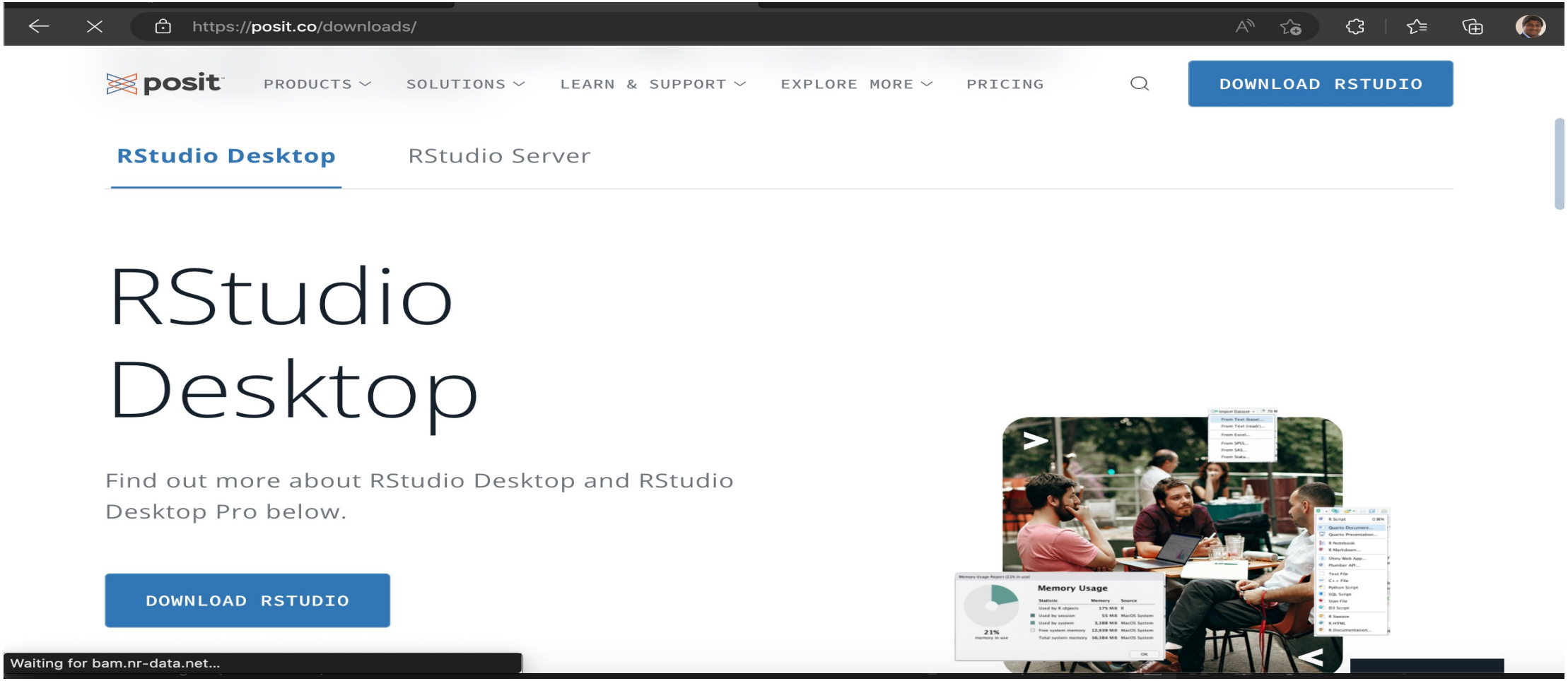
- It can be installed separately as follows:
  - Install.packages(“haven”)
  - load(haven)
  - read_sas(datafile.sas7bdat)
  - read_sav(datafile.sav)
  - read_dta(datafile.dta)

Installing r packages from other sources:

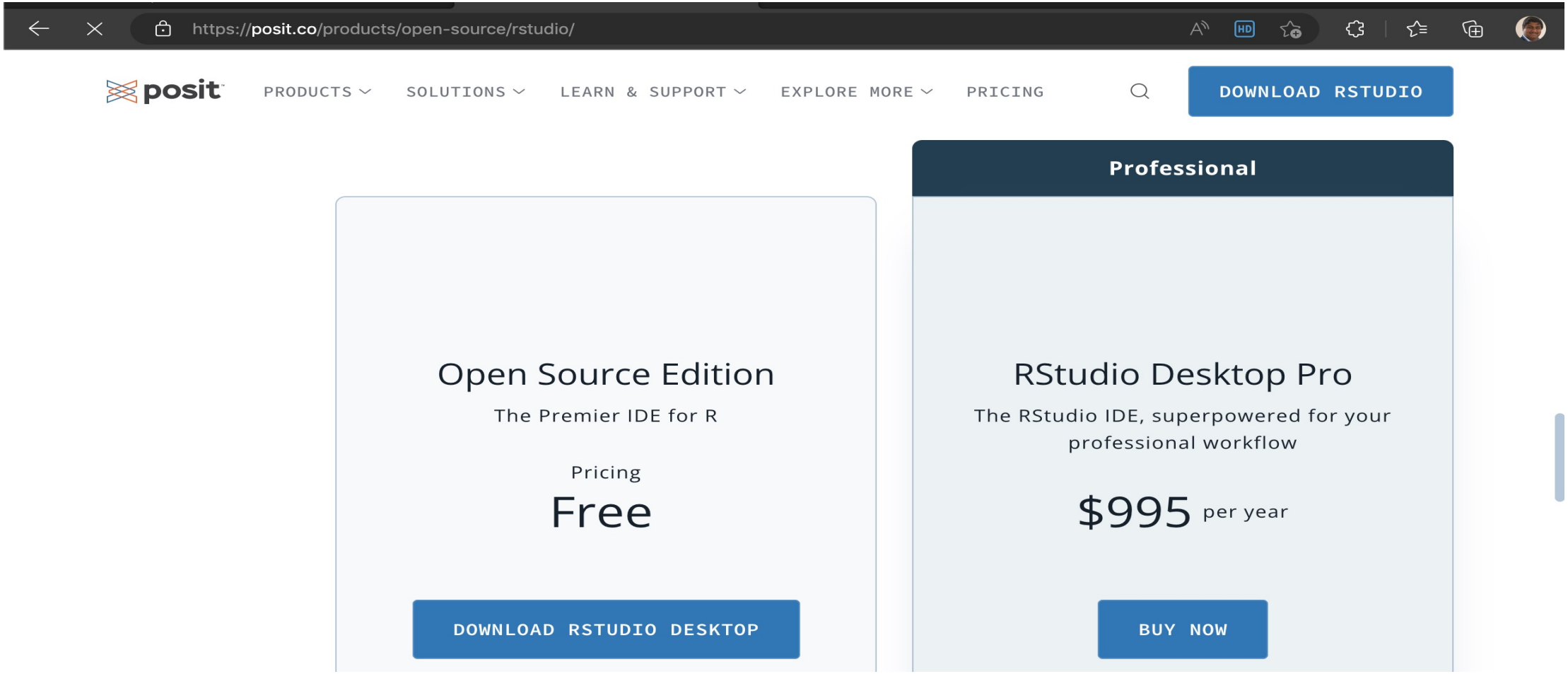
- We can use “GitHub”, which requires “devtools” or “githubinstall” package *a priori*
  - Install_github(“twitter/AnomalyDetection”)
  - githubinstall(“AnomanyDetection”)
  - We can install “latest” packages using this method but with a cost!

- We can use “Bioconductor” repository if we intend to work with Genomics/Bio-Informatics
  - Install Bioconductor Manager first
  - BioCManager::install(c(“GenomicFeature”, “AnnotationDbi”)

R Studio Desktop (Next class)
https://posit.co/downloads



R Studio Desktop IDE (Next class)
https://posit.co/products/open-source/rstudio



Question/Queries?

Thank you!

@shitalbhandary