Statistical Computing with R Masters in Data Science 503 (S1) First Batch, SMS, TU, 2021

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.

Course Description:

- This is an <u>outcome based course</u> 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 (Zoom and Google Classroom):

- Didactic session
- Individual work
- Group 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**
  - Microsoft R open- The Enhanced R Distribution?

- IDE
  - **R Studio**
  - Visual Studio/Visual Studio Code?

- Packages
  - **Base**
  - ***Recommended (tidyverse packages for data sciences)***
  - As per the requirement (Rattle/Weka, etc.)

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
- 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

https://cran.r-project.org

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

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2021-08-10, Kick Things) R-4.1.1.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

What are R and CRAN?

(Simple) Data entry in R:

RGui (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> data <- c(1,2,3,4,5,6,7,8,9)
> data
[1] 1 2 3 4 5 6 7 8 9
> text <-c("a", "b", "c", "d", "e", "f", "g", "h", "i")
> text
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i"
> text2 <-c(a, b, c, d, e, f, g, h, i)
Error: object 'a' not found
> data2 <-cbind(data,text)
> data2
     data text
[1,] "1"  "a"
[2,] "2"  "b"
[3,] "3"  "c"
[4,] "4"  "d"
[5,] "5"  "e"
[6,] "6"  "f"
[7,] "7"  "g"
[8,] "8"  "h"
[9,] "9"  "i"
>
```

c = column vector, we can use r for row vector

Why “Error”?

So, data2 is defined as a matrix with two uni-dimensional arrays i.e. data and text.

type class(data2) in R and check the result.

[1] "matrix" "array"

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.

RGui (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
> M <- matrix(
+ c(1:9),
+ nrow = 3, ncol =3,
+ byrow = TRUE
+ )
> print(M)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
> V <- c(1:12)
> #Multidimentional array
> MDA <- array(V, dim(2,3,))
Error in dim(2, 3, ) : argument 3 is empty
> MDA <- array(V, dim(2,3,2))
Error in dim(2, 3, 2) : 3 arguments passed to 'dim' which requires 1
> MDA <- array(V, dim = (2,3,2))
Error: unexpected ',' in "MDA <- array(V, dim = (2,"
> MDA <- array(V, dim = c(2,3,2))
> print(MDA)
, , 1

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2

     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12

>
```

See how a “function” is used here!

Why “errors” here?

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

Quick Think! (Group work in breakout rooms) Individual assignment at Google Classroom!

- 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:

RGui (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
> 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")),
+    stringsAsFactors = FALSE
+ )
> print(emp.data)
  emp_id emp_name salary start_date
1      1     Rick 623.30 2012-01-01
2      2      Dan 515.20 2013-09-23
3      3 Michelle 611.00 2014-11-15
4      4     Ryan 729.00 2014-05-11
5      5     Gary 843.25 2015-03-27
```

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 data from Data Frame in R:

```
> # 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):

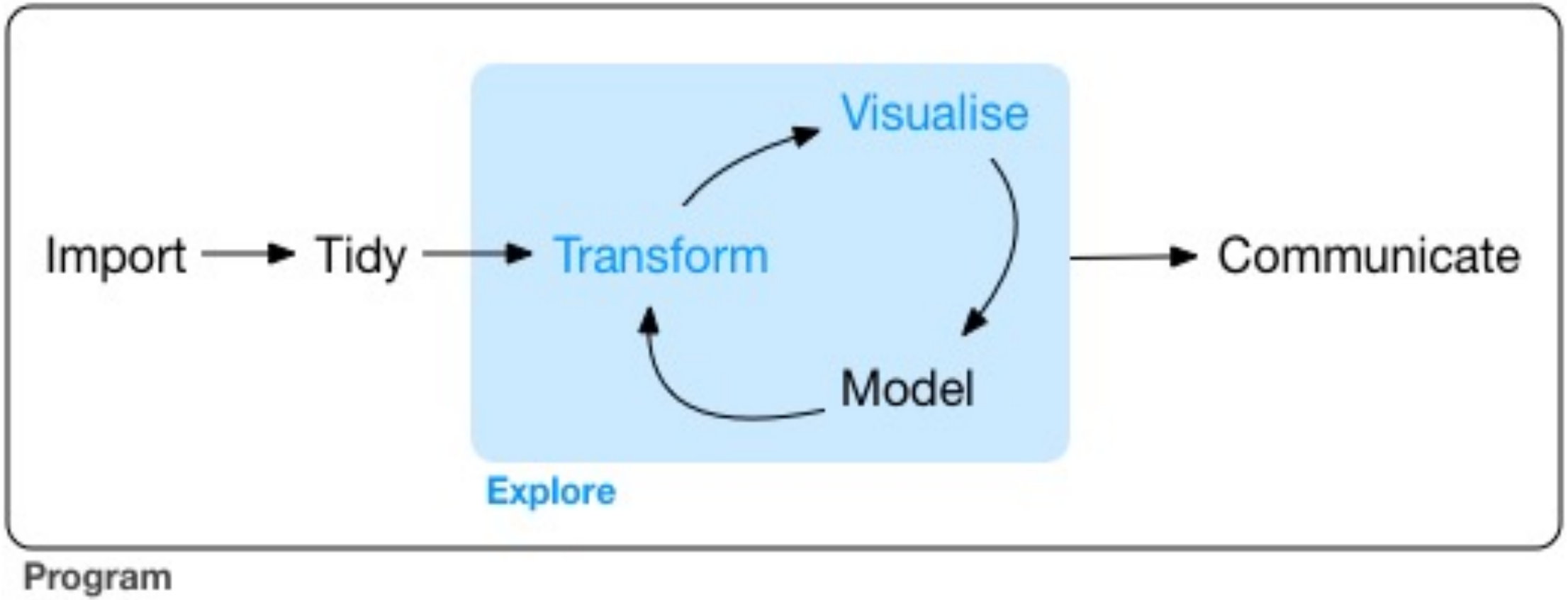
```
> # Bind the two data frames.
> emp.finaldata <- rbind(emp.data,emp.newdata)
> print(emp.finaldata)
  emp_id emp_name salary start_date        dept
1      1     Rick 623.30 2012-01-01          IT
2      2      Dan 515.20 2013-09-23 Operations
3      3 Michelle 611.00 2014-11-15          IT
4      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())

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!

RGui (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
> read.csv("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
28   2/19/2020           0        0                1                0           0         0
29   2/20/2020           0        0                2                1           0         0
30   2/21/2020           0        0                2                0           0         0
31   2/22/2020           0        0                2                0           0         0
32   2/23/2020           0        0                2                0           0         0
```

This worked because the the data file was available in the default working director.

Check it with getwd() in R!

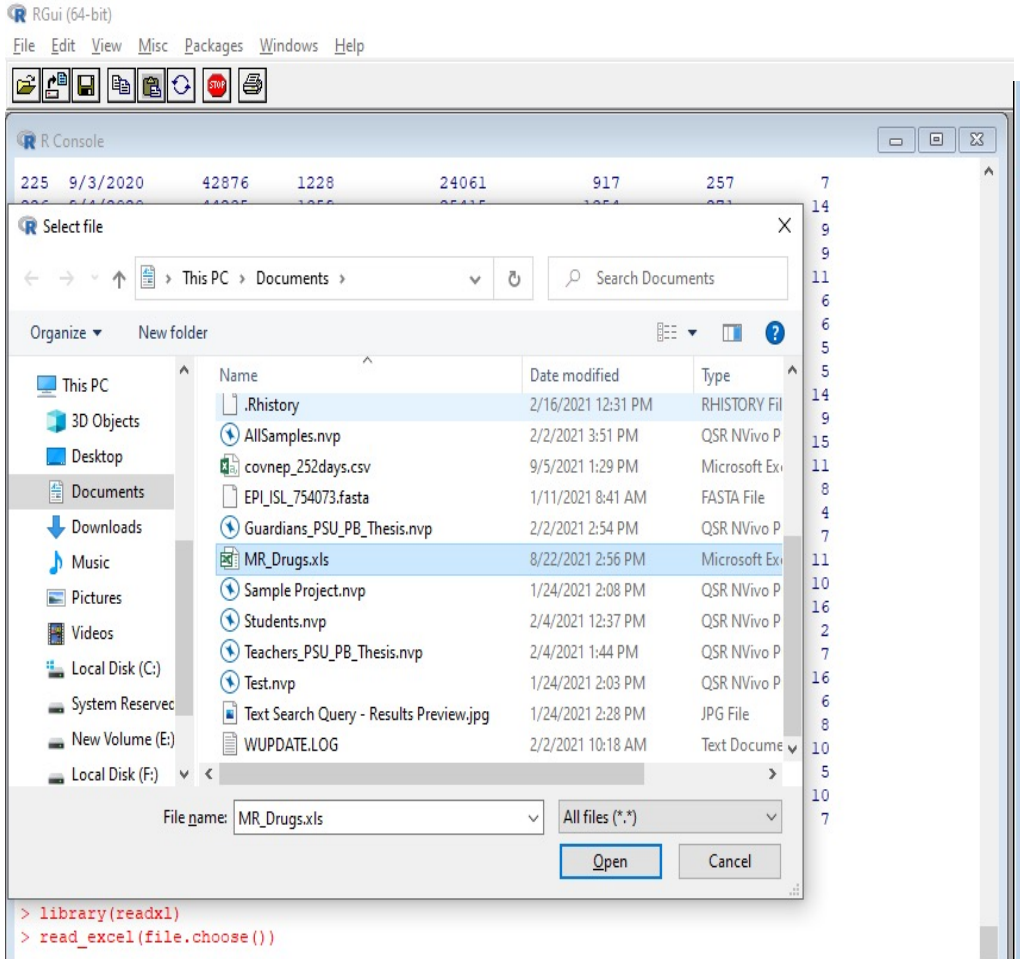
Import data in R: Excel files

- 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 cann’t read excel file with xlsx extensions!

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

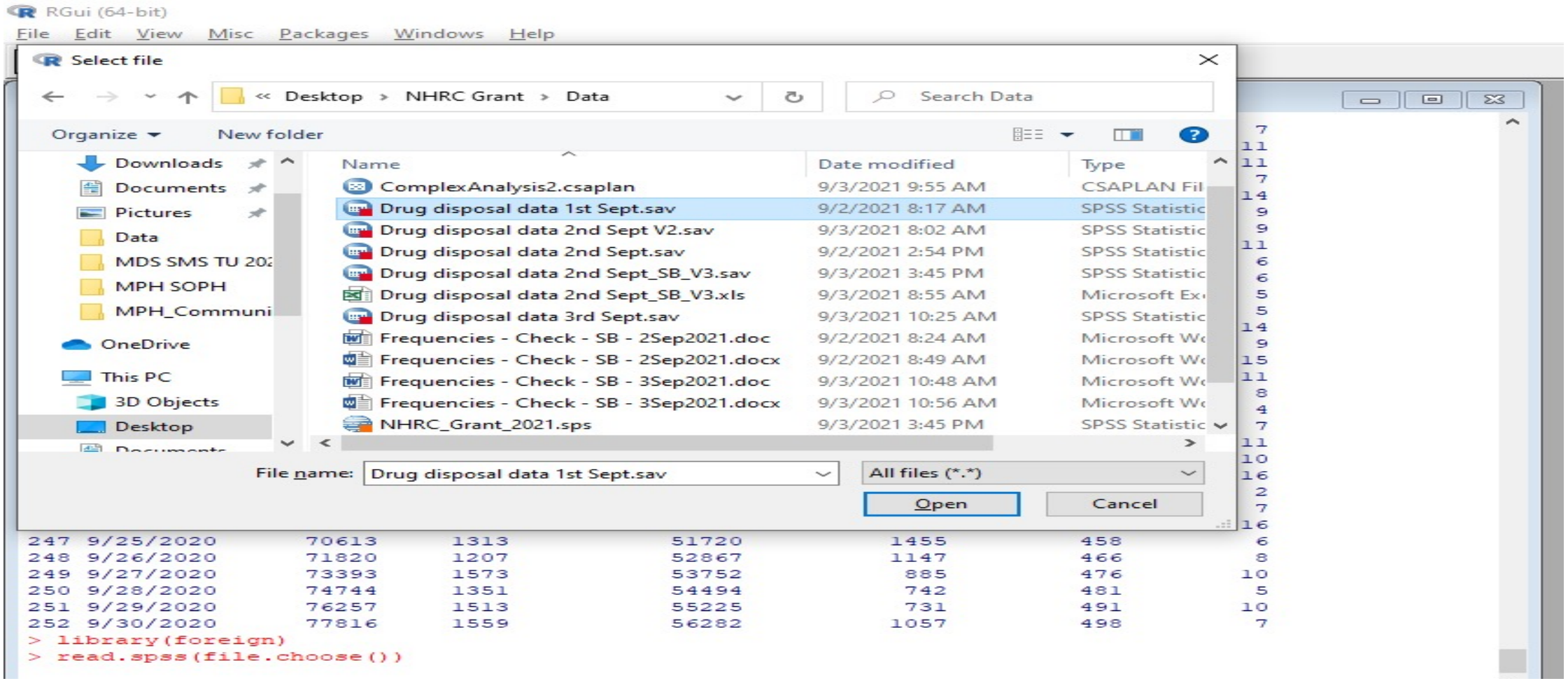



```
> library(readxl)
> read_excel(file.choose())
# A tibble: 972 x 27
      id   sex  city incol inco2 inco3 inco4 inco5 inco6 inco7 pinco1 pinco2 pinco3 pinco4 pinco5
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
 1  1001     2     1     0     0     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:

https://www.tidyverse.org/packages/

Tidyverse

Packages Blog Learn Help Contribute

Tidyverse packages

Installation and use

- Install all the packages in the tidyverse by running `install.packages("tidyverse")`.
- Run `library(tidyverse)` to load the core tidyverse and make it available in your current R session.

Learn more about the tidyverse package at https://tidyverse.tidyverse.org.

Contents

Core “tidyverse” packages

Tidyverse

Packages | Blog | Learn | Help | Contribute

ggplot2

ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details. Go to docs...

dplyr

dplyr provides a grammar of data manipulation, providing a consistent set of verbs that solve the most common data manipulation challenges. Go to docs...

tidyr

tidyr provides a set of functions that help you get to tidy data. Tidy data is data with a consistent form: in brief, every variable goes in a column, and every column is a variable. Go to docs...

readr

readr provides a fast and friendly way to read rectangular data (like csv, tsv, and fwf). It is designed to flexibly parse many types of data found in the wild, while still cleanly failing when data unexpectedly changes. Go to docs...

Contents

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://www.rstudio.com/products/rstudio/download/

R Studio Desktop (Next class onwards)
https://www.rstudio.com/products/rstudio/download/

https://www.rstudio.com/products/rstudio/download/

	RStudio Desktop	RStudio Desktop Pro	RStudio Server	RStudio Workbench
	Open Source License	Commercial License	Open Source License	Commercial License
	Free	$995 /year	Free	$4,975 /year (5 Named Users)
	DOWNLOAD	BUY	DOWNLOAD	BUY
	Learn more	Learn more	Learn more	Evaluation | Learn more
Integrated Tools for R	✓	✓	✓	✓
Priority Support		✓		✓
Access via Web Browser			✓	✓
RStudio Professional Drivers		✓		✓
Connect to RStudio Workbench remotely		✓		
Enterprise Security				✓

Question/Queries?

Thank you!

@shitalbhandary