

Workshop on R Part Two

R Workshop 2019



Introduction

- This is a hands-on workshop on R
- The hands-on learning could be done in a computer lab
- The reference material for this workshop will be a free textbook on basic R
- The free book along with the accompanying data sets can be found at <https://learningstatisticswithr.com/>
- Material has been chosen from the above free book that is most relevant for this workshop



Workshop Outline Part Two

- Note: this workshop is for internal training purposes only. No content in this workshop is intended for re-distribution in any format.
- Next, Chapter 4 of free book (FB)
<https://learningstatisticswithr.com/>

Topic Outline – follow Chapter 4 of Free Book

- 4.1 Using comments
- 4.2 Installing and loading packages
- 4.3 The R workspace
- 4.4 The computer's file system
- 4.5 Loading and saving data with R
- 4.6 More about R variables
- 4.7 Factors in R
- 4.8 Data Frames
- 4.9 Lists
- 4.10 Formulas
- 4.11 Generic Functions
- 4.12 R's Help System

4.1

- Using comments
- Helpful to remind yourself a few months later what you were doing!

```
R Console (64-bit)
File Edit Misc Packages Windows Help

> # Code may be easier to read with comments
> # All code after the comment character is treated
> # as a comment by the R interpreter
>
> # Example
>
> # create the first variable, length of rectangle
> x <- 3.1415
>
> # create the second variable, width of rectangle
> y <- 2.7183
>
> # now multiply them to find the area of the rectangle
> xy <- x*y
>
> # print out the value of the variable xy (area of the rectangle)
> print( xy)
[1] 8.539539
>
>
>
>
> # compare this to:
>
> x <- 3.1415# create the first variable, length of rectangle
> y <- 2.7183# create the second variable, width of rectangle
> xy <- x*y # now multiply them to find the area of the rectangle
> print( xy)# print out the value of the variable xy (area of the rectangle)
[1] 8.539539
>
>
>
> # compare this to:
>
> x <- 3.1415
> y <- 2.7183
> xy <- x*y
> print( xy)
[1] 8.539539
>
> |
```

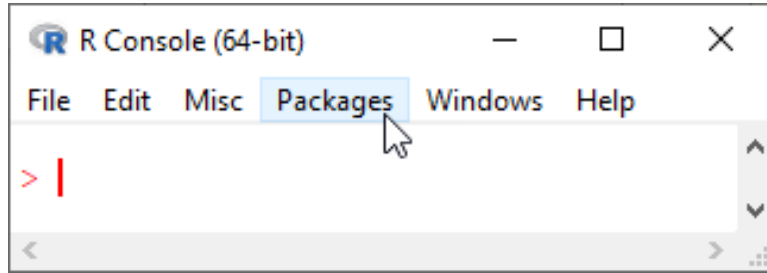
Next, 4.2 R packages

4.2

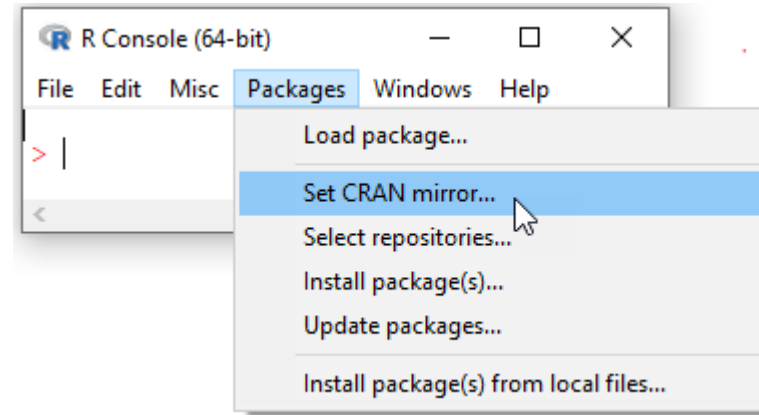
R packages

- R packages make R famous!
- Core packages versus add on packages
- Be careful with add on packages!

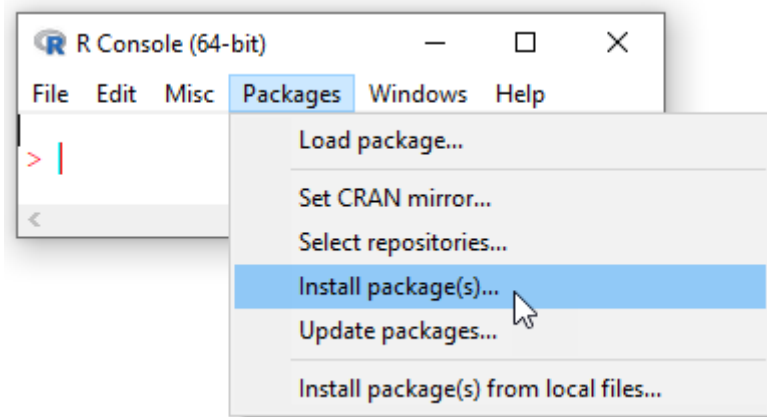
Step 1



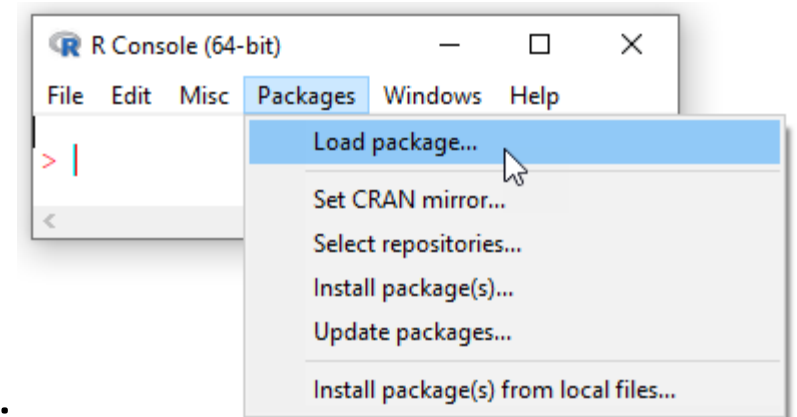
Step 2



Step 3



Step 4



- Core packages come with default R installation.

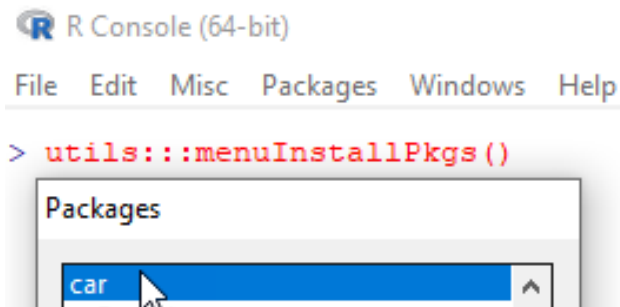
- FB uses extra functionality from three add on packages: lsr, psych, and car. Installing and loading these three packages should let you run all R code in FB.

4.2

R packages continued

- Example: installing and loading and unloading the R package car
- Check for package updates using Packages menu

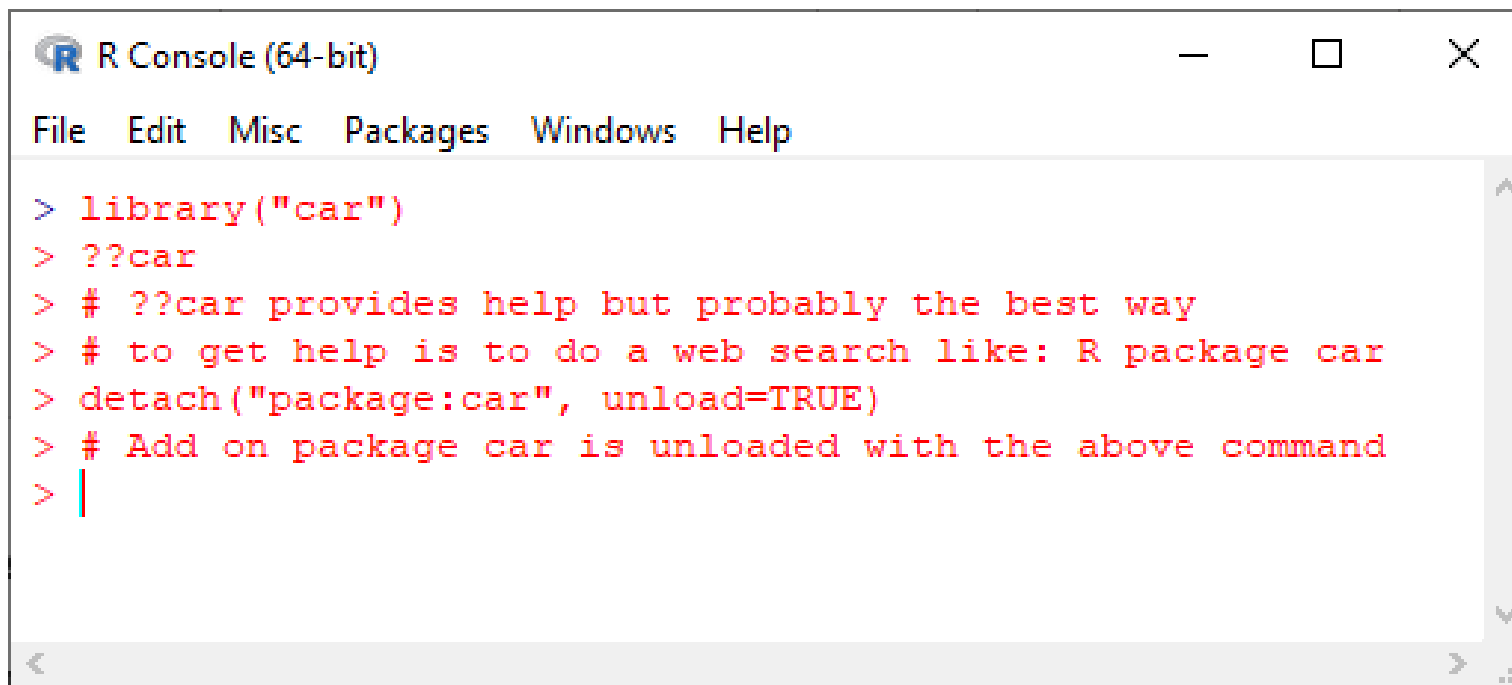
Step 1



```
R Console (64-bit)
File Edit Misc Packages Windows Help
> utils:::menuInstallPkgs ()
```

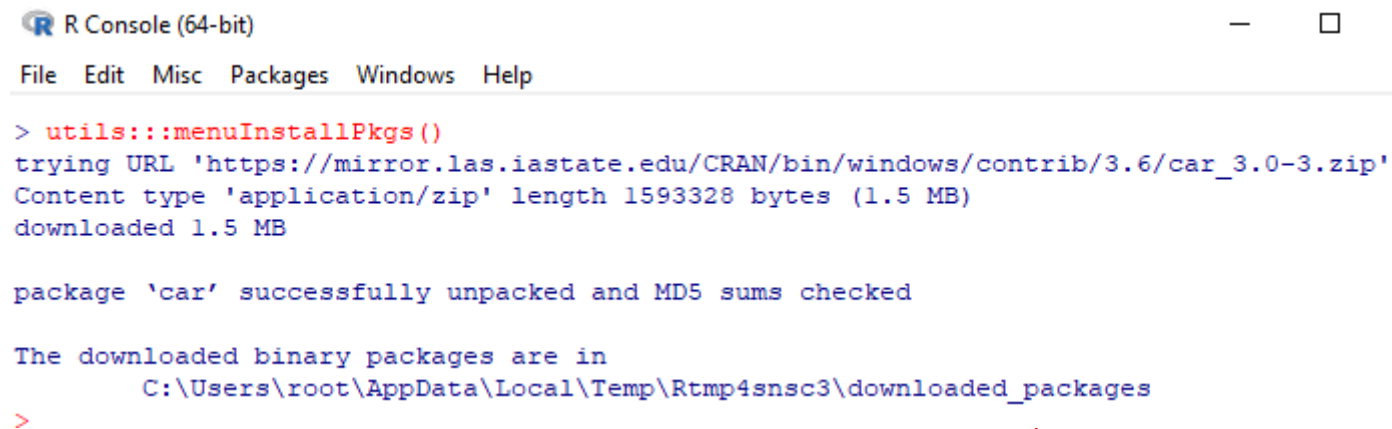
The screenshot shows a dialog box titled "Packages" with a list of available packages. The package "car" is selected and highlighted in blue.

- Load the installed package car by typing in the R command `library()` and unload using the command `detach()`



```
R Console (64-bit)
File Edit Misc Packages Windows Help
> library("car")
> ??car
> # ??car provides help but probably the best way
> # to get help is to do a web search like: R package car
> detach("package:car", unload=TRUE)
> # Add on package car is unloaded with the above command
> |
```

Result (note: Iowa State CRAN Mirror was chosen)



```
R Console (64-bit)
File Edit Misc Packages Windows Help
> utils:::menuInstallPkgs ()
trying URL 'https://mirror.las.iastate.edu/CRAN/bin/windows/contrib/3.6/car_3.0-3.zip'
Content type 'application/zip' length 1593328 bytes (1.5 MB)
downloaded 1.5 MB

package 'car' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\root\AppData\Local\Temp\Rtmp4snsc3\downloaded_packages
>
```

- After loading the car package, the functions that are available in car can be used in your R code, R script, or R program

4.3 – 4.5

- Example of workspace objects:

```
R Console (64-bit)
File Edit Misc Packages Windows Help
> # The objects you create are kept in the R workspace
> # to see the objects in your current workspace use
> # the R function objects(). Note that the workspace will not
> # be saved automatically. If you want to keep workspace for later
> # save workspace
> objects()
 [1] "annual.sales"      "any.sales.this.month"
 [3] "days.per.month"  "february.sales"
 [5] "greeting"         "months"
 [7] "profit"           "sales.by.month"
 [9] "stock.levels"     "var.false"
[11] "var.false.too"    "x"
[13] "X"                "xy"
[15] "y"
> # The above objects were created from R commands given earlier|
```

R workspace, access to file system, and loading data

- Workspace can be saved (Rdata files)
- Need to understand R file system basics load local data files and to run local R scripts (R files)
- One way to get data in to R (load data) is by reading in a CSV file

- Example of file loaded and printed:

```
R Console (64-bit)
File Edit Misc Packages Windows Help
> # When working with files like "mydata.csv" that you want to
> # to learn the current working directory use getwd()
> # put your file "mydata.csv" in the current directory or
> # change the current working directory to where you want wi
> getwd()
[1] "C:/Users/root/Documents"
> # Put file mydata.csv in directory C:/Users/root/Documents $
> myregdata <- read.csv( file = "mydata.csv", header = TRUE )
Warning message:
In read.table(file = file, header = header, sep = sep, quote $
  incomplete final line found by readTableHeader on 'mydata.c$
> print ( myregdata )
  Yvar Xvar
1    3  1.0
2    6  2.0
3    9  2.8
```

- Notepad text, getwd(), and csv file:

```
workspace-file-system-load-data.txt - Notepad
File Edit Format View Help
# When working with files like "mydata.csv" that you want to import it is important
# to learn the current working directory use getwd()
# put your file "mydata.csv" in the current directory or
# change the current working directory to where you want with setwd()
getwd()
# Put file mydata.csv in directory C:/Users/root/Documents then read it in and print with
myregdata <- read.csv( file = "mydata.csv", header = TRUE )
print ( myregdata )
```

```
R Console (64-bit)
File Edit Misc Packages Windows Help
> getwd()
[1] "C:/Users/root/Documents"
> |
```

```
mydata.csv - ...
File Edit Format View Help
"Yvar", "Xvar"
3,1
6,2
9,2.8
```



```
R Console (64-bit)
File Edit Misc Packages Windows Help

> # Inf and -Inf designate the concept of infinity
> # NaN not a number, calculation did not result in a number
> # NA
> # NA stands for not available, similar to missing in other numerical software
> # It is common in large datasets to have missing values, R represents with NA
> # NULL means value does not exist (but it is not missing)
>
> # Giving vector elements names
>
> profit <- c( 3.1, 0.1, -1.4, 1.1 )
> profit
[1] 3.1 0.1 -1.4 1.1
>
> # assigning a vector of labels (character strings) to names(profit)
> names(profit) <- c("Q1","Q2","Q3","Q4")
> profit
  Q1  Q2  Q3  Q4
3.1 0.1 -1.4 1.1
>
>
> # delete labels with the command names(profit) <- NULL
>
> # A short cut to get the same result with this command:
>
> profit <- c( "Q1" = 3.1, "Q2" = 0.1, "Q3" = -1.4, "Q4" = 1.1 )
> profit
  Q1  Q2  Q3  Q4
3.1 0.1 -1.4 1.1
>
>
> #Get value of profit for Q1
> profit["Q1"]
  Q1
3.1
>
```

4.6 More on R variables

```
R Console (64-bit)
File Edit Misc Packages Windows Help

> # To output label names:
> names(profit)
[1] "Q1" "Q2" "Q3" "Q4"
>
> # Types of variables, data type of an R variable
>
> # Some data types in R
> # character (text) data, numeric data, logical data
>
> # Using wrong data type can cause problems
>
> x <- "apples"# x is character
> y <- "oranges" # y is character
> x * y # Error
Error in x * y : non-numeric argument to binary operator
>
>
> x <- "5"# x is character
> y <- "4"# y is character
> x * y # Error
Error in x * y : non-numeric argument to binary operator
>
> # The class() function outputs data type
>
>
> x <- "hello world"# x is text
> class(x)
[1] "character"
>
> x <- TRUE# x is logical
> class(x)
[1] "logical"
>
> x <- 100# x is a number
> class(x)
[1] "numeric"
> |
```

4.6

Next, 4.7 Factors

4.7

Factors represent nominal levels used in statistical analysis

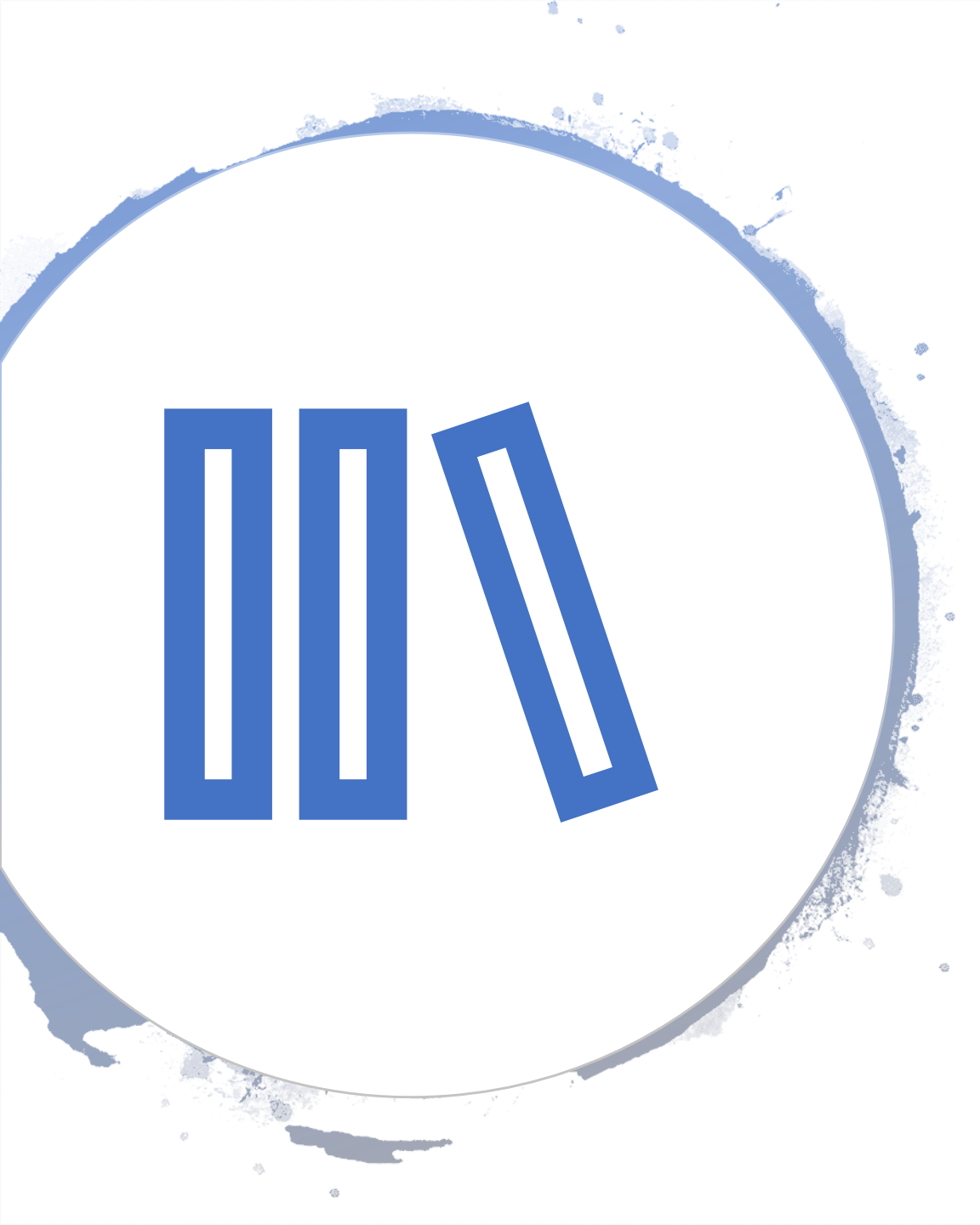
```
R Console (64-bit)
File Edit Misc Packages Windows Help

>
> # Incorporate data that is nominal with factor function in R
> # Example: Response variable, real income of first job after graduation
> # Explanatory variables: major, age, gender, time between graduation
> # and first job, university
> # university can take one of three values UVic, UBC, and SFU
>
> # How to code the variable univ? It will be coded as a factor
>
> # Example
>
> univdata <- c(1,1,1,2,2,2,3,3,3)
>
> # So here univdata is a vector of numerical values
>
> # Now turn univdata into a factor
>
> univ <- as.factor(univdata)
> univ
[1] 1 1 1 2 2 2 3 3 3
Levels: 1 2 3
>
> class(univ)
[1] "factor"
>
> # add labels for clarity
>
> levels(univ) <- c("UVic","UBC","SFU")
> print(univ)
[1] UVic UVic UVic UBC  UBC  UBC  SFU  SFU  SFU
Levels: UVic UBC SFU
>
> |
```



R Workshop Part Two: More...

- More chapter 4 material in accompanying pdf files



Questions?

- Questions about the slides?
- Questions about the free book?
- Questions about implementing what has been covered on a computer?

Next... Additional Material and Examples

- Additional material and examples provided in accompanying pdf files...