MATH 3424 Tutorial 1: Brief Introduction to R

Zhongyuan Lyu

R language

- As decribed by The R Foundation, "R is a language and environment for statistical computing and graphics."
  - Open-source, free.
  - Easily extensible through contributed packages that cover much of modern statistics and data science.
- Download R: https://www.r-project.org
- Reference book: David Dalpiaz, Applied Statistics with R, available at http://daviddalpiaz.github.io/appliedstats

Recommended IDE: RStudio

- RStudio is an "integrated development environment" (IDE) for working with R that simplifies many tasks and makes for a friendlier introduction to R.
  - Nice interface via Rmarkdown for integrating R code with text for creating documents. (This means you must have R installed to use RStudio)
  - Highly recommended!
- Download RStudio from here: https://rstudio.com

R Basics

Define objects/variables

Scalar, vector, matrix, list, dataframe:

```
a <- 10                                    # Scalar
b <- TRUE                                  # Logical/Boolean variable
u <- c(1,2,3)                              # Vector
A <- matrix(1:10, nrow = 5, ncol = 2)      # Matrix
l <- list(u=2, v="abc")                    # List
df <- data.frame(
          ID = 1:10,
          Group = sample(0:1, 10, replace=TRUE),
          Var1 = rnorm(10),
          Var2 = seq(0, 1, length.out=10),
          Var3 = rep(c('a', 'b'), each=5)
  )                                        # Data Frame
```

Useful commands:

```r
typeof() # get the R type or storage mode of an object
help() or ? # get online help about functions in R
help(log) # get online help of function "log()"
?log # get online help of function "log()"
```

Arithmetic Operations

Operation on scalars/vectors

```r
x <- 10; y <- 2;
x + y    # Addition
x - y    # Subtraction
x * y    # Multiplication
x / y    # Division
x ** 2   # Power
x ^ 2   # Power
x %% y   # Modular arithmetic
         # (i.e. find the remainder of x/y)
x %/% y  # Integer division
...
a <- TRUE; b <- FALSE;
!a       # Negation
a&b      # Logical AND
a|b      # Logical OR
```

Operations on matrices

```r
x=1:9
y=9:1
X=matrix(x,3,3)
Y=matrix(y,3,3)
X+Y   ## entry-wise addition of matrices
X-Y   ## entry-wise subtraction
X*Y   ## entry-wise multiplication of matrices
X/Y   ## entry-wise division of matrices
X %*% Y  ## the usual matrix multiplication
t(X)     ## the transpose of a matrix
Z= matrix(c(9, 2, -3, 2, 4, -2, -3, -2, 16), 3, byrow = TRUE)
Z[1,2]  ## the (1,2) entry of Z
Z[1,]   ## the row 1 of Z
Z[,2]   ## the column 2 of Z
solve(Z)   ## the inverse of Z
diag(4)    ## an 4x4 identity matrix
dim(Z)     ## the dimension of a matrix
rowSums(Z)
colSums(Z)
rowMeans(Z)
colMenas(Z)
diag(Z) ## take out the diagonal entries of a matrix
...
```

Mutiple ways of creating vectors

```
x=c(1,2,3,4,5)
x=rep(2,10)                      ##  (2 2 2 2 2 2 2 2 2 2)
x=seq(1,10)                      ##  (1 2 3 4 5 6 7 8 9 10)
x=seq(from=1.5, to=4.2, by=0.3) ##  (1.5 1.8 2.1 2.4 2.7 3.0 3.3 3.6 3.9 4.2)
x=c(42,"Statistics")             ##  ("42","Statistics")
x=rep("A",times=5)               ##  ("A" "A" "A" "A" "A")
length(x)            ## the length of vector x
```

There are other specific examples of operations on vector, matrix, list, dataframe. **See Basic-0.R file on Canvas.**

Manupulate the dataframe

'cars' is a default R dataset that we do not need to install from outside.

```
cars  ## show the data
summary(cars) ## summary of this data frame
names(cars) ## returns the column names
head(cars)  ## returns the first 10 rows
?cars ## get information of this data frame
cars$dist ## the data of the variable "dist", or just the column of the variable "dist"
cars[,2]
cars[[2]]
dim(cars) ## ## the dimension of a dataframe, similar to matrix case
```

Some built-in Functions

Elementary functions

```
exp(x)          # Exponential
sqrt(x)         # Square root
log(x)          # Natural log
sin(x)          # Trigonometric functions
## All functions above can also be applied to a vector/matrix.
cos(pi/2)       # R even contains pi, but only does
                # finite arithmetic
...
```

"Statistical" functions

```
mean(x)         # Average
sum(x)          # Summation
sd(x)           # Standard deviation
var(x)          # Variance
dnorm(x = 3, mean = 2, sd = 5)     # the pdf at x=3 for N(2, 25)
pnorm(q = 3, mean = 2, sd = 5)     # the cdf at x=3 for N(2, 25)
qnorm(p=0.975, mean=2, sd = 5)     # the quantile for probability 0.975
rnorm(n = 10, mean = 2, sd = 5)    # generate a random sample of 10 of N(2, 25)
runif(10, min=0, max=10) # generate a random sample of 10 of Unif(0,10)
sample(1:100, 5, replace = FALSE) # sample 5 random numbers from 1 to 100 without replacement
...
```

Control Flow

if and else

```
x = 1
y = 3
if (x > y) {
  z = x * y
  print("x is larger than y")
} else {
  z = x + 5 * y
  print("x is less than or equal to y")
}
```

```
## [1] "x is less than or equal to y"
```

R also has a special function **ifelse()** which is very useful. It returns one of two specified values based on a conditional statement.

```
ifelse(4 > 3, 1, 0)
```

```
## [1] 1
```

```
fib = c(1, 1, 2, 3, 5, 8)
ifelse(fib > 6, "Foo", "Bar")
```

```
## [1] "Bar" "Bar" "Bar" "Bar" "Bar" "Foo"
```

ifelse() can be applied entry-wisely to a vector. Here, the argument "fib>6" is a logical vector (FALSE,FALSE,FALSE,FALSE,FALSE,TRUE).

For loop

```
x = 11:15
for (i in 1:5) {
  x[i] = x[i] * 2
}
```

However, you should avoid using "for loop" in your coding if possible, as it is "intrinsically" slow in R. For example, you can just use the following code to replace the above one.

```
system.time({
x = 1:100000
for (i in 1:100000) {
  x[i] = x[i] * 2
}
})
```

```
##    user  system elapsed
##   0.019   0.002   0.021
```

```
system.time({
x = 1:100000
x = 2*x
})
```

```
##    user  system elapsed
##       0       0       0
```

Packages

The primary way to install a package ('lme4' for example) is using:

```
install.packages('lme4')
```

To use the package, call:

```
library(lme4)
```

- Some useful packages in R:
  - 'lme4' - Mixed-effects models
  - 'mlr' - Extensible framework for classification, regression, survival analysis and clustering
  - 'dplyr' - Fast data manipulation
  - 'ggplot2' - R's famous package for making beautiful graphics
  - 'knitr' - Easy dynamic report generation in R
  - 'foreach' - Executing the loop in parallel
  - 'glmnet' - Lasso and elastic-net regularized generalized linear models
  - ...

Examples such as hitogram plot of empirical pdf/simulation for CLT can be found in lecture notes.