R CHEAT SHEET

COMPUTATIONAL STATISTICAL
PHYSICS
SUMMER 2017

1. PRELIMINARY

```
# IN R, ALL COMMENTARIES ARE STARTED WITH #.
# HENCE, IN THESE NOTES, ALL SENTENCES STARTED WITH #
# WILL BE COMMENTARIES / ANOTATIONS.
# IF A SENTENCE IS STARTED WITH >, IT CORRESPONDS TO
# AN INSTRUCTION YOU CAN USE IN R.
# I WILL BE WRITING IN CAPS LOCK FOR BETTER UNDERSTANDING.
# HOWEVER, R IS CASE SENSITIVE. TYPICALLY, ALL FUNCTIONS
# SHOULD BE TYPED IN LOWER CASING. IF I UNDERLINE A LETTER,
# IT MEANS IT SHOULD BE UPPER CASED
```

2. To Start

Online Help

```
> help.start()
```

Libraries of Functions

```
> search()   # See currently loaded packages

> library()   # See all packages installed

> .libPaths()   # Get library location

> library(package)   # Load library "package" (if installed)
```

```
# To install a package, just use the package installer
# functionality provided with the software.
```

Clearing Console

Ctrl + L (Windows and Linux machines)

Option + Command + L (Mac OS)

3. DATA ASSIGNMENT AND TYPES

TO ASSIGN A VALUE, STRING OR LOGICAL DATA TO A VARIABLE:

```
> x ← 4    # (OR x = 4)

> y ← "UNIVERSE"

> FLAG1 ← TRUE

> FLAG2 ← FALSE

> x    # PRINTS VALUE OF VARIABLE AFTER ASSIGNMENT
```

CHECKING CLASS OF VARIABLE

```
> CLASS (x)

> TYPE OF x (x)

> IS. DOUBLE (VARIABLE)   # RETURNS TRUE OR FALSE

> IS. INTEGER (VARIABLE)   # SAME AS BEFORE
```

TO ENFORCE TYPE OF VARIABLE

```
> x ← AS. DOUBLE (3,5)

> y ← AS. INTEGER (4)
```

```
# COMPLEX. VARIABLES

> Z ← AS. COMPLEX (-25 + 5i)

# BY DEFAULT, COMPUTATIONS ARE DONE ON REAL NUMBERS.
> SQRT(-1)     # WILL RETURN NOT A NUMBER (NA)

> SQRT( AS. COMPLEX (-1))   # SHOULD RESULT IN +i


# LOGICAL OPERATORS

  <      # SMALLER THAN

  <=     # SMALLER THAN OR EQUAL TO

  >      # LARGER THAN

  >=     # LARGER THAN OR EQUAL TO

 ==      # IS EQUAL TO

 !=      # IS DIFFERENT FROM
```

4. VECTORS

```
> x ← c(10, 5, 3, 6)    # FUNCTION c(x1, x2, x3) ACTS AS A
                        "CONCATENATOR" OF ELEMENTS, INCLUDING
                        STRINGS
```

```
# MOST OPERATIONS ON VECTORS ARE USUALLY PERFORMED ON EACH
# ELEMENT. SEE WHAT HAPPENS WHEN YOU TYPE IN

> x * x

> x + 2

> x ^ 2
```

GENERATING VECTORS

```
> INDEX ← 1 : 20    # SEE WHAT THIS DOES

> μ ← SEQ (FROM = -3, TO = 3, BY = 0.5)  # TRY DIFFERENT
                                           VALUES HERE.

> μ ← SEQ ( -3, 3, LENGTH = 13)

> y←REP (1:4, 4)   # REPEATS INTEGERS 1 TO 4, 4 TIMES

> y ← REP (1:4, c(2, 2, 3, 4))   # REPEATS INTEGERS 1 TO 4,
                                 # WITH EACH INTEGER BEING
                                 # REPEATED THE AMOUNT OF TIMES
                                 # ENCODED IN THE VECTOR c(...)
```

RANDOM NUMBER GENERATORS

> X ← RUNIF (m, X_1, X_2) # GENERATES 10 NUMBERS FROM
THE UNIFORM DISTRIBUTION
BETWEEN X_1 AND X_2 (EXCLUDING
THESE)

> Y ← SAMPLE (X_1 : X_2, m, REPLACE = T)

SAMPLE GENERATES m INTEGERS BETWEEN X_1 AND X_2. IF
REPLACE = T, REPETITIONS CAN OCCUR, OTHERWISE SET TO F.

~~> Z ← RNORM (M, ME~~

> Z ← RNORM (m, MEAN = X_1, SD = X_2) # GENERATES m
NUMBERS FROM THE NORMAL
(GAUSSIAN) DISTRIBUTION,
WITH MEAN = X_1 AND
STANDARD DEVIATION SD = X_2.

> l ← RGAMMA (M, SHAPE, RATE) # GENERATES m NUMBERS
FROM GAMMA DISTRIBUTION.
MUST SPECIFY SHAPE AND
SCALE (BOTH POSITIVE)

```
# ADDITIONAL VECTORS COMMANDS

> LENGTH(x)      # SELF EXPLANATORY

> x[1]           # FIRST ELEMENT OF X

> x[LENGTH(x)]   # FINAL ELEMENT OF X

> x[-j]          # REMOVES j-TH ELEMENT OF X

> x[x>9]         # RETURNS ELEMENTS OF X OBEYING CONDITION

> UNIQUE(x)      # RETURNS ARRAY WITH NON-REPEATED ELEMENTS
                 # OF X

> SUM(x)         # SUMS ELEMENTS OF X

> PROD(x)        # MULTIPLIES ALL ELEMENTS OF X

# AND MANY MORE. ~~REMEMBER,~~ GOOGLE CAN BE YOUR
# BEST FRIEND IN/WHILE PROGRAMMING.
```

5. MATRICES

```
> x ← 1:8

> DIM(x) ← C(2,4)     # FILL A MATRIX (BY ITS COLUMNS)
                      # WITH 2 ROWS AND 4 COLUMNS

> x ← MATRIX(1:8, 2, 4, BYROW = F)   # SHOULD PRODUCE
                                     # SAME RESULT AS BEFORE.
                                     # TRY SETTING BYROW = T

> CBIND(x1, x2)    # BINDS TWO (OR MORE) VECTORS AS
                   # COLUMN VECTORS

> RBIND(x1, x2)    # BINDS TWO (OR MORE) VECTORS AS
                   # ROW VECTORS
```

OPERATIONS ON MATRICES

```
> x ← MATRIX(1:16, 4, 4)    # PRODUCES SQUARE MATRIX 4x4

> y ← 7:10

> x * y      # WHAT DOES THIS DO?

> x * x      # AND THIS?

> x %*% y    # THIS IS PROPER MATRIX MULTIPLICATION.
             # DIMENSIONS OF x AND y MUST BE COMPATIBLE!

> x %*% t(x)    # MULTIPLYING x BY ITS TRANSPOSE
```

6. ARRAYS

# ARRAYS ARE JUST A GENERALIZATION OF VECTORS AND MATRICES.
# ALL BASIC ARITHMETIC OPERATIONS ~~WHICH~~ WHICH APPLY TO MATRICES/VECTORS
# ALSO APPLY TO ARRAYS

```
> X ← ARRAY (C (1:8, 3: 10, 10:20), DIM = C(2,4,3))
  # CREATES MULTI-DIMENSIONAL ARRAY!
```

7. DATA FRAMES

# DATA FRAMES CAN ALSO BE REGARDED AS AN EXTENSION TO MATRICES.
# THESE CAN HAVE COLUMNS OF DIFFERENT DATA TYPES AND ARE
# SEEN AS THE MOST CONVENIENT STRUCTURE FOR DATA IN R.

```
> MTCARS    # PRINTS BUILT-IN DATA FRAME

> ROWNAMES (MTCARS)   ⎫
> NAMES (MTCARS)      ⎪
> MTCARS [1, ]        ⎬  # WHAT DO THESE DO?
> MTCARS [ , 2]       ⎭
```

CREATING DATA FRAMES (TO TRY AT HOME/LATER)

```
# WITH THIS SIMPLE EXAMPLE, YOU WILL CREATE YOUR OWN RANDOMLY
# GENERATED DATA FRAME AND SEE IF YOUR TUTOR IS CAPABLE OF
# TEACHING R OR NOT!

> X ← MATRIX(SAMPLE(C(T, F), 15, REPLACE = T), 5, 3)

> X ← DATA.FRAME(X)     # THIS AUTOMATICALLY CREATES A
                        # DATA FRAME STRUCTURE FOR YOUR MATRIX

> NAMES(X) ← C("PYTHON", "C", "R")   # DEFINE COLUMN NAMES
                                     # OF DATA FRAME

> ROW.NAMES(X) ← C("NELSON", "PETER", "JENNIFER", "LUCA", "MARIA")

> X     # PRINT X

# WHAT WAS THE RESULT? SHOULD I BE YOUR TUTOR FOR COMP.
# STATISTICAL PHYSICS?
```

8. IF / ELSE IN R

```
IF ( TEST- EXPRESSION) {
        ⋮
     STATEMENTS
        ⋮
} ELSE {
   ⋮
OTHER STATEMENTS
   ⋮
}
```

EXAMPLE

```
> X ← SAMPLE (1:10, 1)
> IF (X>5) {
+  PRINT (" X IS LARGER THAN 5 ") } ELSE {
+  PRINT (" X IS SMALLER OR EQUAL TO 5 ") }
```

9. For cycle

```
For (i in Vector) {
      ⋮
    Statements
}
```

Example

```
> x ← 1:100
> For (i in x) {
+ Print (x[i]^2) }
```