# **R** Tutorial

Anup Aprem aaprem@ece.ubc.ca

September 13, 2016

◆□ > 
▲□ >

æ



- Installing R: https://www.r-project.org/
- Recommended to also install R Studio: https://www.rstudio.com/

э



#### Basic element is the column vector

```
# create basic vector
x = c(1,2,3,4,5)
x[1]
## [1] 1
```

The first element of x is 1 given by x[1].

イロト イポト イヨト イヨト

э

### Matrices

Use matrix command to create matrices. Also, matrix operations are similar to vector operations.

```
# create basic matrix
mat=matrix(data=c(9,2,3,4,5,6),ncol=3,nrow=2)
# Show the matrix
mat
       [,1] [,2] [,3]
##
## [1,]
          9
            3
                    5
          2
## [2,]
               4
                    6
# Second row of matrix
mat[2,]
## [1] 2 4 6
```

(日) (得) (물) (물) (물)

# Matrices (Contd.)

```
# create basic matrix
mat=matrix(data=c(9,2,3,4,5,6),ncol=3,nrow=2)
# Third coloumn of matrix
mat[,3]
### [1] 5 6
# Selecting multiple coloumn
mat[,1:2]
### [,1] [,2]
### [1,] 9 3
### [2,] 2 4
```

(日) (圖) (문) (문) (문)

## Vector/Matrix Operations

Vector addition and subtraction

```
# create vector
x = c(0,1,2); y = c(1,2,3)
# Vector addition
x + y
## [1] 1 3 5
# Vector subtraction: x - y
```

#### Dimension

```
# create matrix
mat = matrix(1:12,nrow = 3, ncol = 4)
# Row Dimension
dim(mat)[1]
```

```
## [1] 3
```

```
# Column dimension: dim(mat)[2]
```

Vector/Matrix Operations (Contd.)

which() function one wishes to find the indices of elements that have a certain characteristic.

```
# create vector
y = c(1,2,3,4,5)
# Show index of even numbers
which(y %% 2 == 0)
## [1] 2 4
# which allows complicated logical expressions
# Select elements from y which are less than 5
# but greater than or equal to 3
which((y >= 3) & (y < 5))</pre>
```

## [1] 3 4

イロト イポト イヨト イヨト

Vector/Matrix Operations Contd.

Combined selections for matrices

```
# create matrix
mat <- matrix(1:12, 3, 4, byrow=TRUE)</pre>
# Select all columns starting with a number less than 3
mycols = mat[1,] < 3
mycols
## [1] TRUE TRUE FALSE FALSE
# Show selected coloumns
mat[ , mycols]
        [,1] [,2]
##
## [1,]
       1 2
## [2,] 5 6
## [3,] 9 10
```

3

イロト イポト イヨト イヨト

### Vector/Matrix Operations Contd.

#### Transpose of Matrix

```
# create matrix
mat <- matrix(1:12, 3, 4, byrow=TRUE)</pre>
# Transpose using t operator
t(mat)
       [,1] [,2] [,3]
##
  [1,]
##
          1
            5
                 9
  [2,]
          2 6 10
##
        3 7 11
## [3,]
## [4,]
          4 8 12
```

イロト イポト イヨト イヨト

### Vector/Matrix Operations Contd.

#### Binding vectors to create a matrix

Binding column wise

<i># Create matrix</i>				
<pre>mat &lt;-cbind(c(1,2), c(3,4)</pre>				
mat				
##		[,1]	[,2]	
##	[1,]	1	3	
##	[2,]	2	4	

#### Binding row wise

```
# Create matrix
mat <-rbind(c(1,2), c(3,4))
mat
## [,1] [,2]
## [1,] 1 2
## [2,] 3 4</pre>
```

э

## Vector/Matrix Operations Contd.

#### Matrix Multiplication

```
# Create 2 matrices
mat1 <-cbind(c(1,0), c(0,1)) # Identity Matrix</pre>
mat2 <- matrix(1:4, 2, 2, byrow=TRUE)</pre>
# Element wise multiplication
mat1 * mat2
       [,1] [,2]
##
## [1,] 1 0
## [2,] 0 4
# Matrix Multiplication
mat1 %*% mat2
##
        [,1] [,2]
## [1,]
       1
                2
                4
## [2,]
           3
```

э

イロト イポト イヨト イヨト

## The If-Else control

Syntax: if (test-expression) { if-commands } else { else-commands } Could be used without the else part.

```
x = -5
if(x > 0){
    print("Non-negative number")
} else {
    print("Negative number")
}
## [1] "Negative number"
```

The For Loop

Use the for loop if you want to do the same task a specific number of times. Syntax: for (counter in vector) {commands}

```
for(i in 1:5) {
    print(i^2)
}
### [1] 1
### [1] 4
## [1] 9
### [1] 16
## [1] 25
```

э

イロト イポト イヨト イヨト

### Plots

Use plot command for simple plots<sup>1 2</sup>.



<sup>1</sup>For advanced plotting use ggplot2 package

<sup>2</sup>Make sure your figures look like this while submitting your assignments. (a) = -2

#### Plots contd.

```
# Create a sequence from 0 to 4pi using 0.01 increment
t = seq(0, 4*pi, 0.01)
s=sin(t)
c=cos(t)
# Create a line plot using plot command
plot(t,s,type='l',col='red',xlab="time",ylab="Amplitude")
# Use lines instead of plot to overlay
lines(t,c,type='l',col='blue')
# Switch on grid
grid(nx = 20, ny = 20)
# Use title command for title and axis labels
title(main="Sine and Cosine wave")
# Use legend for distingushing multiple plots
legend("topright", legend=c("Sine","Cosine"), col=c("red","blue"),lwd=1)
```

## Printing plots to pdf

Print plots to pdf<sup>3</sup>.

pdf("HW1\_plot.pdf")
# Plot commands here
dev.off()

<sup>3</sup>Use pdf of the plots you generate in your assignment while submitting.  $4 \equiv 3 = -9$   $\leq 0$ 

### Histogram

A histogram  $\!\!\!\!^4$  is a visual representation of the distribution of a dataset

```
x = c(1,2,3,3,3,4,4,4,4,5,5,5,5,5)
hist(x)
```



<sup>4</sup>Useful in the second part of the course

A B A B A
 A
 B
 A
 A
 B
 A
 A
 B
 A
 A
 B
 A
 A
 B
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A

э

< ∃ >

### Writing your own functions

- Functions can be written both inline or in separate files.
- Use the source command to read functions written in a separate file.

3

```
fun1 = function(arg1, arg2 )
{
w = arg1 ^ 2
return(arg2 + w)
}
fun1(arg1 = 3, arg2 = 5)
## [1] 14
```

# Installing packages

- R has a large number of packages<sup>5</sup>.
- Use install.packages command to install from CRAN.
- You can also install from source.

# Installing ggplot2
install.packages('ggplot2')

<sup>5</sup>This is part of the reason R is so widely popular



- Help about a function use the help() command.
- Use the example() command to get examples of a function.
- Additional packages from CRAN.

э

# Submitting assignments

Please submit a print out of your code<sup>6</sup> and simulation results<sup>7</sup>. In addition, submit your code at https://dropitto.me/STAT\_357\_HW1(password: stat357hw1) (for HW1).

While submitting code, please ensure to do the following

- Use .R or .m extension.
- Name the file as HWx\_QnNo.m (HWx\_QnNo.R). For example, the code for Question 8 of Homework 1 should be named as HW1\_8.m (HW1\_8.R)
- Please zip all the files and rename it to StudentNo\_HW1.zip. (Do not use anything else other than zip)
- Write comments! You can get most of the credit if your logic is correct.

<sup>6</sup>Printing out your code is optional if you submit the code online <sup>7</sup>Submitting a print out of the simulation results is mandatory;⇒ < ≥ > < ≥ > <

