Basic R

Byteflow Dynamics

9/24/2017

Basic Syntax of R

Open an R script file and type the following

```
a <- 2 + 3 + 4
a
```

```
## [1] 9
```

- **<-** is used to define variables
- Mathematical operators like +, -, /, * can be used

Exercise: We sold 55 units yesterday, and 77 units today. Calculate the percent difference.

```
(77 - 55) / 55 * 100
```

```
## [1] 40
```

- **c** for combine

```
a <- c(1, 2, 3)
a
```

```
## [1] 1 2 3
```

This is the same as

```
a <- 1:3
a
```

```
## [1] 1 2 3
```

To access each element, use [ ]

```
a[1] + a[3]
```

```
## [1] 4
```

Exercise: Calculate the product of the second and third elements of a.

```
a[2] * a[3]
```

```
## [1] 6
```

Data Types

- Numeric

```
N <- 24
class(N)
```

```
## [1] "numeric"
```

- Character

```
S <- "new york"
class(S)
```

```
## [1] "character"
```

- Logical

```
L <- TRUE # or FALSE
class(L)
```

```
## [1] "logical"
```

To convert data types

```
as.character(L)
```

```
## [1] "TRUE"
```

- Factor
  - analogous to an enumerated type in other languages
  - useful in statistics, model fitting, etc
  - but not suitable for data manipulation (which is a big part of what we do)
  - R tries to convert characters to factors frequently

Data frame

- tabular data object
- each column is a list of vectors of equal length
- each column can be a different data type

```
N <- c(2, 3, 5)
S <- c("aa", "bb", "cc")
L <- c(TRUE, FALSE, TRUE)
df <- data.frame(N, S, L)
df
```

```
##   N  S     L
## 1 2 aa  TRUE
## 2 3 bb FALSE
## 3 5 cc  TRUE
```

To access each column

```
df[,1]
```

```
## [1] 2 3 5
```

```
df[[2]]
```

```
## [1] aa bb cc
## Levels: aa bb cc
```

```
df[,c(1,3)]
```

```
##   N     L
## 1 2  TRUE
## 2 3 FALSE
## 3 5  TRUE
```

```
df$S
```

```
## [1] aa bb cc
## Levels: aa bb cc
```

To access each row

```
df[1,]
```

```
##   N  S    L
## 1 2 aa TRUE
```

```
df[c(2:3),]
```

```
##   N  S     L
## 2 3 bb FALSE
## 3 5 cc  TRUE
```

To access each value

```
df[2,2]
```

```
## [1] bb
## Levels: aa bb cc
```

```
df[2,2:3]
```

```
##    S     L
## 2 bb FALSE
```

```
df$L[3]
```

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

Combine data frames

Combine by columns: cbind

```
df1 <- data.frame(N =  7:9, S = c("x1","x2","x3"), L = c(FALSE, TRUE, TRUE))
df1
```

```
##   N  S     L
## 1 7 x1 FALSE
## 2 8 x2  TRUE
## 3 9 x3  TRUE
```

```
df2 <- df

df3 <- cbind(df1, df2)
df3
```

```
##   N  S     L N  S     L
## 1 7 x1 FALSE 2 aa  TRUE
## 2 8 x2  TRUE 3 bb FALSE
## 3 9 x3  TRUE 5 cc  TRUE
```

Data frames must have equal number of rows!

Combine by rows: rbind

```
df4 <- rbind(df1, df2)
df4
```

```
##   N  S     L
## 1 7 x1 FALSE
## 2 8 x2  TRUE
## 3 9 x3  TRUE
## 4 2 aa  TRUE
## 5 3 bb FALSE
## 6 5 cc  TRUE
```

Column lengths and names must be equal!

Logical Operators

- & - and
- | - or
- ! - not
- == - equal to
- inequality operators <, >, <=, >=

```
x & y (x and y)
x | y (x or y)
x != y (x is not equal to y)
!x (not x)
```

Example 1: Print subset of df4 where L is not FALSE

```
df4[df4$L != FALSE,]
```

```
##   N  S    L
## 2 8 x2 TRUE
## 3 9 x3 TRUE
## 4 2 aa TRUE
## 6 5 cc TRUE
```

Example 2: Subset of df4 where L is TRUE and N is greater than 3

```
df4[(df4$L == TRUE) & (df4$N > 3),]
```

```
##   N  S    L
## 2 8 x2 TRUE
## 3 9 x3 TRUE
## 6 5 cc TRUE
```

Exercise: Subset df4 based on these conditions: L is not TRUE or N is smaller than 5

```
df4[(df4$N < 5) | (df4$L != TRUE),]
```

```
##   N  S     L
## 1 7 x1 FALSE
## 4 2 aa  TRUE
## 5 3 bb FALSE
```

Read and write data

Read the data files

```
lab <- read.csv('lab_grades.csv', stringsAsFactors = FALSE)
lec <- read.csv('lecture_grades.csv', stringsAsFactors = FALSE)
```

Check the datasets

```
head(lab)
```

```
##   X Last.Name First.Name Lab.1 Lab.2 Lab.3 Lab.4 Lab.5 Lab.6 Lab.7 Lab.8
## 1 1         A          T    NA     8    10    NA    NA     5    NA    NA
## 2 2         B          M    NA    NA    NA    NA    NA    NA    NA    NA
## 3 3         B          J    10     9     8    10    10    10    10    10
## 4 4         B          B    10     9     8    10    10    10    10    10
## 5 5         C          M    10     9     8    NA    10    10    10    10
## 6 6         C          S    10     9     9    10     8     8    10    10
##   Lab.9 Lab.10 Lab.11 Lab.12 Lab.13 Lab.14
## 1    NA     10     NA     NA     NA     10
## 2    NA     NA     NA     NA     NA     NA
## 3    10     10     NA     10   10.0     10
## 4    10     10     10     10    8.5     NA
## 5    NA     10     NA     10    9.0     NA
## 6    10     10     NA     10    9.0     NA
```

```
head(lec)
```

```
##   X Last.Name First.Name Quiz.Ch.2 HW.ch.1 Quiz.Ch.3 HW.ch.4 HW.ch.5
## 1 1         A          T         7      NA       6.5       6      10
## 2 2         B          M        NA      NA        NA      NA      NA
## 3 3         B          J         9       9       8.8       8       7
## 4 4         B          B         8      NA       6.8       8       7
## 5 5         C          M         8      10       9.0       7       9
## 6 6         C          S         8       8       6.3       8      10
##   Midterm HW.Ch.6 HW.Ch.10 Extra.Credit Final.exam
## 1      40       1      6.0           22         39
## 2      NA      NA       NA           NA         NA
## 3      72      10      8.0           40         87
## 4      68       9      7.5           30         69
## 5      81      10       NA           NA         67
## 6      48      10      6.0           35         81
```

What are those NAs?? Replace them with 0s.

```
lab[is.na(lab)] <- 0
lec[is.na(lec)] <- 0
```

```
head(lab)
```

```
##   X Last.Name First.Name Lab.1 Lab.2 Lab.3 Lab.4 Lab.5 Lab.6 Lab.7 Lab.8
## 1 1         A          T     0     8    10     0     0     5     0     0
## 2 2         B          M     0     0     0     0     0     0     0     0
## 3 3         B          J    10     9     8    10    10    10    10    10
## 4 4         B          B    10     9     8    10    10    10    10    10
## 5 5         C          M    10     9     8     0    10    10    10    10
## 6 6         C          S    10     9     9    10     8     8    10    10
##   Lab.9 Lab.10 Lab.11 Lab.12 Lab.13 Lab.14
## 1     0     10      0      0    0.0     10
## 2     0      0      0      0    0.0      0
## 3    10     10      0     10   10.0     10
## 4    10     10     10     10    8.5      0
## 5     0     10      0     10    9.0      0
## 6    10     10      0     10    9.0      0
```

```r
head(lec)
```

```
##   X Last.Name First.Name Quiz.Ch.2 HW.ch.1 Quiz.Ch.3 HW.ch.4 HW.ch.5
## 1 1         A          T         7       0       6.5       6      10
## 2 2         B          M         0       0       0.0       0       0
## 3 3         B          J         9       9       8.8       8       7
## 4 4         B          B         8       0       6.8       8       7
## 5 5         C          M         8      10       9.0       7       9
## 6 6         C          S         8       8       6.3       8      10
##   Midterm HW.Ch.6 HW.Ch.10 Extra.Credit Final.exam
## 1      40       1      6.0           22         39
## 2       0       0      0.0            0          0
## 3      72      10      8.0           40         87
## 4      68       9      7.5           30         69
## 5      81      10      0.0            0         67
## 6      48      10      6.0           35         81
```

To save a data frame as a csv file, use write.csv.

Exercise: Save df4 as a csv file

```r
write.csv(df4, "df4.csv")
```