1 - Intro to R

Data Structures

Karsten Maurer, Eric Hare, and Susan VanderPlas

Iowa State University

Data Frames

- Data Frames are the work horse of R objects
- Structured by rows and columns and can be indexed
- Each column is a specified variable type
- Columns names can be used to index a variable
- Advice for naming variable applys to editing columns names
- Can be specified by grouping vectors of equal length as columns

Data Frame Indexing

- Elements indexed similar to a vector using `[ ]`
- `df[i,j]` will select the element in the i^{th} row and j^{th} column
- `df[ ,j]` will select the entire j^{th} column and treat it as a vector
- `df[i ,]` will select the entire i^{th} row and treat it as a vector
- Logical vectors can be used in place of i and j used to subset the row and columns

Adding a New Variable to Data Frames

- Create a new vector that is the same length as other columns
- Append new column to the data frame using the $ operator
- The new data frame column will adopt the name of the vector

Data Frame Demo

- Demo using a statistical classic: Edgar Anderson's Iris Data
- Follow along with the R script named 4-DataStructures.R

Your Turn

- Make a data frame with column 1: 1,2,3,4,5,6 and column 2:a,b,a,b,a,b
- Select only rows with value "a" in column 2 using logical vector
- `mtcars` is a built in data set like `iris`: read the values from row 4 by indexing

Lists

- Lists are a structured collection of R objects
- R objects in a list need not be the same type
- Create lists using the `list()` function
- Lists indexed using double square brackets [[ ]] to select an object

List Example

```
# Create a list including a vector and a matrix
mylist <- list( matrix(letters[1:10], nrow=2, ncol=5) , seq(0, 49, by=7
mylist
```

```
## [[1]]
##      [,1] [,2] [,3] [,4] [,5]
## [1,] "a"  "c"  "e"  "g"  "i"
## [2,] "b"  "d"  "f"  "h"  "j"
##
## [[2]]
## [1]  0  7 14 21 28 35 42 49
```

```
# Use index to select second object in list, this will return the vecto

mylist[[2]]
```

```
## [1]  0  7 14 21 28 35 42 49
```

Your Turn

- Create a list containing a vector and a 2x3 data frame
- Use indexing to select the data frame from your list
- Use further indexing to select the first row from the data frame in your list

Examining Objects

- `head(x)` - View top 6 rows of a data frame
- `tail(x)` - View bottom 6 rows of a data frame
- `summary(x)` - Summary statistics
- `str(x)` - View structure of object
- `dim(x)` - View Dimensions of object
- `length(x)` - Returns the length of a vector

Examining Objects Example

```
# Examine the top 2 rows of the iris data set from built in data packag
head(iris, 2)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
```

```
# How big is this data set?
dim(iris)
```

```
## [1] 150   5
```

```
# What structure does the data set have?
str(iris)
```

```
## 'data.frame': 150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1
```

Your Turn

- View the top 6 rows of mtcars data
- What type of object is the mtcars data set?
- How many rows are in iris data set? (try finding this using dim or indexing + length)
- Summarize the values in each column in iris data set

Working with output from a function

- Can save output from a function as an object
- Object is generally a list of output objects
- Can pull off items from the output for further computing
- Examine object using functions like `str(x)`

Output Object Demo

- Demo of saving t-test output as an object

Your Turn

- Pull the p-value from the t-test of a difference between Sepal Lengths of setosa and versicolor species from the Iris data

Importing Data

- First need to tell R where the data is saved using `setwd()`
- Data read in using R functions such as:
  - `read.table()` for reading in .txt files
  - `read.csv()` for reading in .csv files
- Assign the data to new R object when reading in the file

Importing Data Demo

- Demo of creating a csv file and loading it into R

Your Turn

- Make 5 rows of data in an excel spreadsheet and save it as a .txt file
- Import this new .txt file into R with read.table