Introduction to Data Frames

INFO 201

Today’s Objectives

Feel confident with working with ***functions*** and ***vectors***

Familiarize yourself with the ***list*** object type

Begin working with R’s primary 2D storage type, the **data frame**

Learn how to **load** R's datasets, and **read** and **write** .csv data

Function and Vector Review

module 7 exercise-3

Lists

Lists

A variable type to store multiple elements (similar to vectors)

Some important differences:

- Sequence of elements which can be **different types**
- Element can be **added** or **removed** from lists

Created using the **list** keyword:

```
my.list <- list(1, '2', TRUE)
```

Retrieving List Information

It's best practice to **name** list elements (possible, but less common with vectors):

```
person1 <- list(name = "Miriam", favorite.band = "Red Baraat")
```

Two options to retrieve information by **name**

```
person1$name # returns "Miriam", with `$` notation

person1[['name']] # returns "Miriam", with double bracket notation
```

Possible, but much less clear to reference information by index

```
person1[[1]] # returns "Miriam"
```

Single v.s. Double Brackets

When using lists:

Single brackets will return a list of specified elements (not their values)

```
person1['name'] # returns the equivalent of list("Miriam")
```

Double brackets will return the desired value *itself*

```
person1[['name']] # returns "Miriam"
```

More on this topic

Module-8 exercise-1

Data Frames

Data Frames

Provide a two-dimensional (row, column) data structure

Most of the time, you'll be working with data frames

Actually a **list** of **vectors** with additional abilities

Each **column** is a **vector** (and therefore, all column elements are of the same **type**)

Creating Data Frames

Use the **data.frame** function to combine multiple vectors into a data frame:

```
# Vector of heights
height <- 58:62

# Vector of weights
weight <- c(115, 117, 120, 123, 126)

# Combine the vectors into a data.frame
my.data <- data.frame(height, weight)
```

Function	Result
nrow(my.data.frame)	Number of rows in the data frame
ncol(my.data.frame)	Number of columns in the data frame
dim(my.data.frame)	Dimensions (rows, columns) in the data frame
colnames(my.data.frame)	Column names of the data frame
rownames(my.data.frame)	Row names of the data frame
head(my.data.frame)	Prints the top subset of rows of the data frame
tail(my.data.frame)	Prints the last subset of rows of the data frame
View(my.data.frame)	Opens up the RStudio data frame viewer (only in RStudio)

Module-8 exercise-2

Accessing Data in Data Frames

Data frames are just lists!

```
# Combine vectors into a data.frame
my.data <- data.frame(height, weight)

# Retrieve a data.frame of only the height column
my.data$height # using the $
my.data['height'] # using column names

# Retrieve the values of a column (in a vector)
my.data[['height']] # returns a vector of the values
```

Syntax	Example	Result
data[row.num, col.num]	data[2,3]	Retrieve element in the second row, third column
data[row.num, col.name]	data[2,'height']	Retrieve second element in the height column
data[row.name, col.name]	data['steve','height']	Retrieve the height of row named steve
data[row.num,]	data[2,]	Retrieve all columns and only the second row
data[row.nums,]	data[2:4,]	Retrieve all columns and the second through fourth rows
data[, col.name]	data[,'height']	Retrieve all rows and only the height column
data[boolean vector, col.name]	data[data$height > 100 ,'height']	Retrieve rows where height is > 100, and only the name column

Module-8 exercise-3

Loading and Reading Data

Loading Data

R comes with a variety of pre-defined datasets

```
data() # view a list datasets, in RStudio
```

To load a particular dataset, pass in it's name to the data function

```
data('Seatbelts') # With or without quotes is OK.
```

You may need to coerce the value to a data frame element

```
# The Seatbelts object is available, but not a data frame

my.data <- as.data.frame(Seatbelts)
```

Loading Data

One of the most common storage formats is ***comma separated values*** files (.csv)

They can be created/edited in excel, but only store the basic data (no formatting)

```
a, b, c,

1, 2, 3
```

Easily read into R with the read.csv function

```
# Read in a data file: give the relative path to the file

my.data <- read.csv('data/filename.csv')
```

Module-8 exercise-4

Upcoming...

By Thursday: Be familiar with **module 8**

Due Tuesday, 10/18 (***before class***): a3-using-data