6: Lists

Introduction to R

This assignment consists of six parts:

- Intro to Basics
- Vectors
- Matrices
- Factors
- Data frames
- Lists (this document)

If you haven't already, download the answer sheet for this document (**Note:** right-click and save the file with the `.Rmd` extension to your `hw02` folder.)

After you complete each exercise, push the R Notebook to your remote repo. See Part 0 for instructions. Do *not* push this document.

6.1 Lists, why would you need them?

Congratulations! At this point in the course you are already familiar with:

- **Vectors** (one dimensional array): can hold numeric, character or logical values. The elements in a vector all have the same data type.
- **Matrices** (two dimensional array): can hold numeric, character or logical values. The elements in a matrix all have the same data type.
- **Data frames** (two-dimensional objects): can hold numeric, character or logical values. Within a column all elements have the same data type, but different columns can be of different data type.

Pretty sweet for an R newbie, right? ;-)

A list in R is similar to your to-do list at work or school: the different items on that list most likely differ in length, characteristic, and type of activity that has to be done.

A list in R allows you to gather a variety of objects under one name (that is, the name of the list) in an ordered way. These objects can be matrices, vectors, data frames, even other lists, etc. It is not even required that these objects are related to each other in any way.

You could say that a list is some kind super data type: you can store practically any piece of information in it!

6.2 Creating a list

Let us create our first list! To construct a list you use the function `list()`:

```
my_list <- list(comp1, comp2, ...)
```

The arguments to the `list` function are the list components. Remember, these components can be matrices, vectors, other lists, ...

Instructions

Construct a list, named `my_list`, that contains the variables `my_vector`, `my_matrix` and `my_df` as list components.

```
# Vector with numerics from 1 up to 10
my_vector <- 1:10

# Matrix with numerics from 1 up to 9
my_matrix <- matrix(1:9, ncol = 3)

# First 10 elements of the built-in data frame mtcars
my_df <- mtcars[1:10,]

# Construct list with these different elements:


#
```

6.3 Creating a named list

Well done, you're on a roll!

Just like on your to-do list, you want to avoid not knowing or remembering what the components of your list stand for. That is why you should give names to them:

```
my_list <- list(name1 = your_comp1,
                name2 = your_comp2)
```

This creates a list with components that are named `name1`, `name2`, and so on. If you want to name your lists after you've created them, you can use the `names()` function as you did with vectors. The following commands are fully equivalent to the assignment above:

```
my_list <- list(your_comp1, your_comp2)
names(my_list) <- c("name1", "name2")
```

Instructions

- Change the code of the previous exercise (see below) by adding names to the components. Use for `my_vector` the name `vec`, for `my_matrix` the name `mat` and for `my_df` the name `df`.
- Print out my_list so you can inspect the output.

```
# Vector with numerics from 1 up to 10
my_vector <- 1:10

# Matrix with numerics from 1 up to 9
my_matrix <- matrix(1:9, ncol = 3)

# First 10 elements of the built-in data frame mtcars
my_df <- mtcars[1:10,]

# Adapt list() call to give the components names
my_list <- list(my_vector, my_matrix, my_df)

# Print out my_list


#
```

6.4 Creating a named list (2)

Being a huge movie fan (remember your job at LucasFilms), you decide to start storing information on good movies with the help of lists.

Start by creating a list for the movie "The Shining". We have already created the variables mov, act and rev in your R workspace. Feel free to check them out in the console.

Instructions

Complete the code below to create `shining_list`; it contains three elements:

- moviename: a character string with the movie title (stored in `mov`)
- actors: a vector with the main actors' names (stored in `act`)
- reviews: a data frame that contains some reviews (stored in `rev`)

Do not forget to name the list components accordingly. The names to use are are `moviename`, `actors` and `reviews`).

```
# The variables mov, act and rev are available
mov <- "The Shining"
act <- c("Jack Nicholson",
         "Shelley Duvall",
         "Danny Lloyd",
         "Scatman Crothers",
         "Barry Nelson" )

scores <- c(4.5, 4, 5.0)
sources <- c("IMDb1", "IMDb2", "IMDb3")
comments <- c("Best Horror Film I Have Ever Seen",
              "A truly brilliant and scary film from Stanley Kubrick",
              "A masterpiece of psychological horror")
rev <- data.frame(scores, sources, comments)

# Finish the code to build shining_list


#
```

6.5 Selecting elements from a list

Your list will often be built out of numerous elements and components. Therefore, getting a single element, multiple elements, or a component out of it is not always straightforward.

One way to select a component is using the numbered position of that component. For example, to "grab" the first component of `shining_list` you type

```
shining_list[[1]]
```

A quick way to check this out is typing it in the console. *Important to remember:* to select elements from vectors, you use single square brackets: `[ ]`. Don't mix them up!

You can also refer to the names of the components, with `[[ ]]` or with the `$` sign. Both will select the data frame representing the reviews:

```
shining_list[["reviews"]]
shining_list$reviews
```

Besides selecting components, you often need to select specific elements out of these components. For example, with `shining_list[[2]][1]` you select from the second component, actors (`shining_list[[2]]`), the first element (`[1]`). When you type this in the console, you will see the answer is Jack Nicholson.

Instructions

- Select from `shining_list` the vector representing the actors. Simply print out this vector.
- Select from `shining_list` the second element in the vector representing the actors. Do a printout like before.

```
# shining_list is already pre-loaded in the workspace

# Print out the vector representing the actors


# Print the second element of the vector representing the actors


#
```

6.6 Adding more movie information to the list

Being proud of your first list, you shared it with the members of your movie hobby club. However, one of the senior members, a guy named M. McDowell, noted that you forgot to add the release year. Given your ambitions to become next year's president of the club, you decide to add this information to the list.

To conveniently add elements to lists you can use the `c()` function, that you also used to build vectors:

```
ext_list <- c(my_list , my_val)
```

This will simply extend the original list, `my_list`, with the component `my_val`. This component gets appended to the end of the list. If you want to give the new list item a name, you just add the name as you did before:

```
ext_list <- c(my_list, my_name = my_val)
```

Instructions

- Complete the code below such that an item named `year` is added to the `shining_list` with the value `1980`. Assign the result to `shining_list_full`.
- Finally, have a look at the structure of `shining_list_full` with the `str()` function.

```
# shining_list, the list containing movie name, actors and reviews, is pre-loaded in the workspace

# We forgot something; add the year to shining_list


# Have a look at shining_list_full


#
```

You now know how to make, use, and modify the basic structures in R: vectors, matrices, data frames, and lists using base R commands.

Upcoming exercises will expand on your new knowledge base. It will also make some of the tasks (e.g., sorting), much easier.