Lists in R: Takeaways

Syntax

CREATING LISTS

- Create a list:

```
new_list <- list("data scientist", c(50000,40000), "programming experience")
```

- Assign names to list objects:

```
names(new_list) <- c("job title", "salaries", "requirements")
```

INDEXING LISTS

- Return a list of selected elements:

```
new_list[1]

new_list["job title"]

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

- Return a single element:

```
new_list[[1]]

new_list[["job title"]]

new_list$"job title"
```

- Return a value contained in a list element:

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

MANIPULATING LISTS

- Modify List Elements

```
new_list[[1]] <- "junior data scientist"

new_list[[c(2,1)]] <- 40000
```

- Add Elements to Lists

```
new_list[[4]] <- c("healthcare", "vacation")

new_list[["benefits"]] <- c("healthcare", "vacation")
```

- Remove Object from Lists

```
new_list[-3]
```

- Combine Multiple Lists

```
new_list_2 <- c(new_list, new_list_1)
```

- Create a List of Lists (Nested List)

```
new_list_3 <- list(new_list, new_list_1)
```

Concepts

- In this course we will learn the following data structures:
    - Vector: one-dimensional structure for storing values of SAME TYPE.
    - Matrix: two-dimensional structure for storing values of SAME TYPE.
    - **Lists: multi-dimensional stucture for storing values of ANY DATA TYPE/OBJECT.**
    - Dataframe: multi-dimensional structure for storing values of ANY DATA TYPE/OBJECT like datasets.

- In R, lists are specialized vectors that can contain multiple objects. The objects may consist of different data structures, including single data elements, vectors, and matrices.
- Storing objects in lists allows you to make use of R's features for performing the same operation on each object in your list.
- Lists of lists contain multiple lists as objects. Each list contained in a nested list may, in turn, contain objects of any data structure or type.

Resources

- Documentation on Lists in R