I have prepared a lecture note on R Lists for you. Here is the content:

What is a List?

A list is a data structure in R that can store a collection of objects. The objects in a list can be of different types, such as numbers, strings, and vectors.

Why use Lists?

There are several reasons why you might want to use lists in R. For example, lists can be used to:

* Store a collection of related data.
* Pass multiple arguments to a function.
* Create a data structure that is more flexible than a vector or matrix.

How to create a List in R

To create a list, you use the list() function. The syntax for creating a list is:

Code snippet

list(

object1,

object2,

...

)

Here,

* object1 is the first object in the list.
* object2 is the second object in the list.
* ...

For example, the following code creates a list called my\_list that contains the numbers 1, 2, and 3:

Code snippet

my\_list <- list(1, 2, 3)

How to Access List Elements in R

There are different ways to access elements of a list. We can use [ , [[ ]], or $ to access specific element of a list in R. For example,

* To access the first element of a list, we can use the following code:

Code snippet

my\_list[1]

* To access the second element of a list, we can use the following code:

Code snippet

my\_list[2]

* To access the third element of a list, we can use the following code:

Code snippet

my\_list[3]

How to Modify a List Element in R

To change a list element, we can simply reassign a new value to the specific index. For example,

Code snippet

my\_list[1] <- 10

This will change the first element of the list from 1 to 10.

How to Add Items to R List

We use the append() function to add an item at the end of the list. For example,

Code snippet

append(my\_list, 4)

This will add the number 4 to the end of the list.

How to Remove Items From a List in R

R allows us to remove items for a list. We first access elements using a list index and add negative sign - to indicate we want to delete the item. For example,

Code snippet

my\_list[-1]

This will remove the first element of the list.

Length of R List

In R, we can use the length() function to find the number of elements present inside the list. For example,

Code snippet

length(my\_list)

This will return the number of elements in the list.

Loop Over a List

In R, we can also loop through each element of the list using the for loop. For example,

Code snippet

for (i in my\_list) {

print(i)

}

This will print each element of the list.

Check if Element Exists in R List

In R, we use the %in% operator to check if the specified element is present in the list or not and returns a boolean value.

TRUE - if specified element is present in the list FALSE - if specified element is not present in the list For example,

Code snippet

"1" %in% my\_list # TRUE

"5" %in% my\_list # FALSE

Here,

"1" is present in my\_list, so the method returns TRUE "5" is not present in my\_list, so the method returns FALSE

I hope this lecture note is helpful. Please let me know if you have any questions.

—----------xxxxxx—------

Here is a lecture note on R Lists:

A list is a data structure in R that can store a collection of objects of different types.

To create a list, you use the list() function. For example,

Code snippet

list1 <- list(24, "Sabby", 5.4, "Nepal")

This creates a list with four elements: an integer, a string, a float, and a string.

You can access elements of a list using their index numbers. For example, to access the first element of list1, you would use:

Code snippet

list1[1]

This would return the integer 24.

You can also access elements of a list using their names. For example, to access the element named "Sabby", you would use:

Code snippet

list1["Sabby"]

This would return the string "Sabby".

You can add elements to a list using the append() function. For example, to add the integer 3.14 to list1, you would use:

Code snippet

append(list1, 3.14)

This would create a new list with five elements: 24, "Sabby", 5.4, "Nepal", and 3.14.

You can remove elements from a list using the [[ operator. For example, to remove the first element of list1, you would use:

Code snippet

list1[[1]] <- NULL

This would create a new list with three elements: "Sabby", 5.4, and "Nepal".

You can also remove elements from a list using the rm() function. For example, to remove the element named "Sabby", you would use:

Code snippet

rm(list1["Sabby"])

This would create a new list with three elements: 24, 5.4, and "Nepal".

You can use the length() function to find the number of elements in a list. For example, to find the number of elements in list1, you would use:

Code snippet

length(list1)

This would return the number 4.

You can use the for loop to iterate over the elements of a list. For example, to print the elements of list1, you would use:

Code snippet

for (i in 1:length(list1)) {

print(list1[[i]])

}

This would print the following output:

Code snippet

[1] 24

[1] "Sabby"

[1] 5.4

[1] "Nepal"

You can use the ifelse() function to check if an element exists in a list. For example, to check if the integer 3.14 exists in list1, you would use:

Code snippet

ifelse(3.14 %in% list1, "Yes", "No")

This would return the string "Yes".

I hope this lecture note is helpful. Please let me know if you have any questions.

Xxxx—-----xxx—-------xx-

Here is a lecture note on R Lists:

What is a List?

A list is a collection of objects in R. It is similar to an array, but it can contain objects of different types.

How to Create a List in R

To create a list in R, you use the list() function. The syntax for creating a list is:

Code snippet

list(object1, object2, ...)

For example, the following code creates a list named my\_list that contains the numbers 1, 2, and 3:

Code snippet

my\_list <- list(1, 2, 3)

How to Access List Elements in R

To access elements of a list in R, you use the [ operator. The syntax for accessing a list element is:

Code snippet

my\_list[[index]]

For example, the following code accesses the second element of the my\_list list:

Code snippet

my\_list[[2]]

How to Add Elements to a List in R

To add elements to a list in R, you use the append() function. The syntax for adding an element to a list is:

Code snippet

append(list, object)

For example, the following code adds the number 4 to the my\_list list:

Code snippet

append(my\_list, 4)

How to Remove Elements from a List in R

To remove elements from a list in R, you use the remove() function. The syntax for removing an element from a list is:

Code snippet

remove(list, index)

For example, the following code removes the second element of the my\_list list:

Code snippet

remove(my\_list, 2)

How to Sort a List in R

To sort a list in R, you use the sort() function. The syntax for sorting a list is:

Code snippet

sort(list)

For example, the following code sorts the my\_list list in ascending order:

Code snippet

sort(my\_list)

How to Loop Over a List in R

To loop over a list in R, you use the for loop. The syntax for looping over a list is:

Code snippet

for (i in list) {

# do something with i

}

For example, the following code prints each element of the my\_list list:

Code snippet

for (i in my\_list) {

print(i)

}

How to Check if an Element Exists in a List in R

To check if an element exists in a list in R, you use the %in% operator. The syntax for checking if an element exists in a list is:

Code snippet

object %in% list

For example, the following code checks if the number 3 exists in the my\_list list:

Code snippet

3 %in% my\_list

How to Get the Length of a List in R

To get the length of a list in R, you use the length() function. The syntax for getting the length of a list is:

Code snippet

length(list)

For example, the following code gets the length of the my\_list list:

Code snippet

length(my\_list)

I hope this lecture note is helpful. Please let me know if you have any questions.