CADS Module-Notes on Important Operators and Functions in R

A. How to Use Pipes

Use of the pipe operator "%>%" in R can be very helpful.

The basic syntax is as follows:

df %>%

Do-this-operation %>%

Then-do-this-operation %>%

Then-do-this-operation…..

The pipe operator "%>%" simply acts of the data frame "df" and pass the result on to the next operation…

I just suggest an improvement as follows:

df_piped <- df %>%

Do-this-operation %>%

Then-do-this-operation %>%

Then-do-this-operation…..

Caution: This creates a new data frame that captures all the transformations to the original data frame. "df_piped" is now available to be used. "df_piped" could be a data frame or some summary result from the lines of code processed. So, it is not always a data frame.

The pipe is especially useful in working with grouped data.

Guided Practice 1:

Install.packages()

library(dplyr)

head(mtcars) # View the first six rows of dataset mtcars

Let us see the value of pipe operator to summarize grouped data without extracting each group separately.

mtcars %>%

```
    group_by (cyl) %>%
    summarize (mean_mpg=mean(mpg))
```

Please see the inserted link for more examples.

https://www.statology.org/?s=Pipe+in+r

Guided Practice 2a.

Create the Example data:

```
Df1 <- data.frame(value= c(1,4,5,2,5,4),
    team = rep(LETTERS[1:3], each=2))
Df1 # Print data frame.
```

Table 1

	value	team
1	1	A
2	4	A
3	5	B
4	2	B
5	5	C
6	4	C

Calculate Mean by Group and as New Colum using the ave() Function

```
Df2 <- Df1
Df2$gr_mean <- ave(Df2$value, Df2$team) # Calculate mean for each group
Df2 # Print new data frame
```

Table 2

	value	team	gr_mean
1	1	A	2.5
2	4	A	2.5
3	5	B	3.5
4	2	B	3.5
5	5	C	4.5
6	4	C	4.5

Guided Practice 2b:

Calculate Mean by Group and add new column using the pipe, group by, mutate function.

```
Intall.packages()
Library(dplyr)
Df3 <- Df1 %>%
        group_by(team) %>%
        mutate(gr_mean = mean(value)) %>%
                as.data.frame
Df3  # Print new data frame.
```

Table 3

	value	team	gr_mean
1	1	A	2.5
2	4	A	2.5
3	5	B	3.5
4	2	B	3.5
5	5	C	4.5
6	4	C	4.5

B. CADS Module-Notes on "%in%" Operator

The operator can be described as 'belong' or 'is in' or 'is an element of' what is on the it's right.

As a logical operator, it can also be used in the negative sense as shown below.

Guided Practice 3a: Produce a new vector that contains elements of data2 that belong in data1.

A simple example shows how to use this on a vector:

```
data1 <- c(3,5,7,7,14,19,22,25) # define vector
data2 <- c(1,2,3,4,5) # define another vector.
data3 <- data1[data1 %in% data2]
print(data3) # Should give answer as 3 5
```

Please see more examples for data frames below.

This gives us another way to modify data frame or extract a subset of a data frame.

Guided Practice 3b: Produce new vector that contains element of data 2 that DO NOT belong in data 1:

```
data4 <- data1[!(data1 %in% data2)]

print(data4) # This should give 7,7,14,19,22,25
```

Please see more examples for data frames below.

This offers us efficient ways to modify data frames.

https://www.statology.org/in-operator-in-r/

More Links

1. Quick Introduction to ggplot2
    a. https://bookdown.org/agrogankaylor/quick-intro-to-ggplot2/quick-intro-to-ggplot2.html
2. How to count observations by Group in R
    a. https://www.statology.org/count-by-group-r/
3. How to Use the transform function in R
    a. https://www.statology.org/transform-function-in-r/
4. How to Use the relocate function in R
    a. https://www.statology.org/dplyr-relocate/
5. How to Arrange Rows in R
    a. https://www.statology.org/arrange-rows-r/
6. How to Use the transmute function in R
    a. https://www.statology.org/r-dplyr-transmute/
7. How to Count Number Rows in R
    a. https://www.statology.org/number-of-rows-in-r/
8. How to Use Mutate to Create New Variables in R
    a. https://www.statology.org/how-to-use-mutate-to-create-new-variables-in-r/