

Data Manipulation using dplyr

dplyr is a package in R that provides a grammar of data manipulation, enabling you to easily manipulate data in a data frame or tibble. Here are some commonly used functions in dplyr:

`select()`: Selects specific columns from a data frame or tibble.

`filter()`: Filters rows based on a specified condition.

`mutate()`: Creates new columns based on calculations or transformations of existing columns.

`arrange()`: Sorts rows based on one or more columns.

`group_by()`: Groups the data by one or more columns.

`summarize()`: Calculates summary statistics for each group.

`distinct()`: Removes duplicate rows based on a specific column or columns.

`rename()`: Renames specific columns in a data frame or tibble.

`left_join()`, `right_join()`, `inner_join()`, `full_join()`: Joins two data frames or tibbles based on a common column or columns.

`case_when()`: Creates conditional statements to generate new columns.

`if_else()`: Creates a conditional statement based on a logical expression.

Illustration on mtcars data

Here's an illustration of how to use dplyr to manipulate the mtcars data:

Load the dplyr package by running `library(dplyr)`.

Create a tibble from the mtcars data frame:

R Copy code # Create a tibble from the mtcars data frame `mtcars_tbl <- as_tibble(mtcars)`
Select specific columns from the mtcars tibble using `select()` function. For example, select the mpg, cyl, and hp columns: R Copy code # Select specific columns from the mtcars tibble
`selected_cols <- select(mtcars_tbl, mpg, cyl, hp)` Filter rows based on a condition using the `filter()` function. For example, filter the mtcars tibble to only include cars with a mpg greater than or equal to 20: R Copy code # Filter rows based on a condition `filtered_tbl`

`<- filter(mtcars_tbl, mpg >= 20)` Create new columns based on existing columns using the `mutate()` function. For example, add a new column called `kmpl` that contains the `mpg` column converted to kilometers per liter: R Copy code `# Create new columns based on existing columns`
`mtcars_tbl <- mutate(mtcars_tbl, kmpl = mpg * 0.425144)` Sort rows based on one or more columns using the `arrange()` function. For example, sort the `mtcars` tibble by descending `mpg`: R Copy code `# Sort rows based on one or more columns arranged`
`mtcars_tbl <- arrange(mtcars_tbl, desc(mpg))` Group the data by one or more columns using the `group_by()` function and calculate summary statistics for each group using the `summarise()` function. For example, group the `mtcars` tibble by the `cyl` column and calculate the mean `mpg` for each group: R Copy code `# Group the data by one or more columns and calculate summary statistics`
`for each group`
`grouped_tbl <- group_by(mtcars_tbl, cyl) %>% summarise(mean_mpg = mean(mpg))`