Session 2: Introduction to dplyr

Wali Reheman

2024-09-10

Introduction to dplyr

In this class, we will explore the **dplyr** package for data manipulation in R. You will learn how to use its key functions such as `select()`, `filter()`, `arrange()`, and `mutate()`. We will also cover advanced topics like using `across()` for applying functions to multiple columns, grouping and summarizing data, and joining datasets.

Loading Libraries and Dataset

We begin by loading the **tidyverse** package (which includes dplyr) and using the built-in **mtcars** dataset. If you haven't installed **tidyverse**, run:

```r
install.packages("tidyverse")
```

Now, load the library and view the first few rows of the dataset:

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Part 1: Selecting and Renaming Columns

1.1 Select Specific Columns

The `select()` function extracts specific columns from a dataset. For example, to select only `mpg`, `hp`, and `cyl`:

```r
mtcars_selected <- mtcars %>%
  select(mpg, hp, cyl)
head(mtcars_selected)
```

```
##                    mpg  hp cyl
## Mazda RX4         21.0 110   6
## Mazda RX4 Wag     21.0 110   6
## Datsun 710        22.8  93   4
## Hornet 4 Drive    21.4 110   6
## Hornet Sportabout 18.7 175   8
## Valiant           18.1 105   6
```

Task 1:
Exercise: Select columns `wt`, `qsec`, and `gear` from the mtcars dataset.

```r
# Your code here
```

1.2 Renaming Columns

Use `rename()` to change column names without altering the data structure. For example, to rename `mpg` to `Miles_Per_Gallon`:

```r
mtcars_renamed <- mtcars %>%
  rename(Miles_Per_Gallon = mpg)
head(mtcars_renamed)
```

```
##                   Miles_Per_Gallon cyl disp  hp drat    wt  qsec vs am gear
## Mazda RX4                     21.0   6  160 110 3.90 2.620 16.46  0  1    4
## Mazda RX4 Wag                 21.0   6  160 110 3.90 2.875 17.02  0  1    4
## Datsun 710                    22.8   4  108  93 3.85 2.320 18.61  1  1    4
## Hornet 4 Drive                21.4   6  258 110 3.08 3.215 19.44  1  0    3
## Hornet Sportabout             18.7   8  360 175 3.15 3.440 17.02  0  0    3
## Valiant                       18.1   6  225 105 2.76 3.460 20.22  1  0    3
##                   carb
## Mazda RX4            4
## Mazda RX4 Wag        4
## Datsun 710           1
## Hornet 4 Drive       1
## Hornet Sportabout    2
## Valiant              1
```

Task 2:
Exercise: Rename the `hp` column to `Horsepower`.

```
# Your code here
```

Part 2: Filtering Rows

2.1 Filter Based on One Condition

Use `filter()` to select rows meeting a condition. For example, filtering cars with more than 6 cylinders:

```
mtcars_filtered <- mtcars %>%
  filter(cyl > 6)
head(mtcars_filtered)
```

```
##                      mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.44 17.02  0  0    3    2
## Duster 360          14.3   8 360.0 245 3.21 3.57 15.84  0  0    3    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.07 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.73 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.78 18.00  0  0    3    3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.25 17.98  0  0    3    4
```

2.2 Filter Based on Multiple Conditions

Combine conditions using logical operators (`&` for AND, `|` for OR). For example, filtering cars with more than 6 cylinders and more than 100 horsepower:

```
mtcars_filtered_advanced <- mtcars %>%
  filter(cyl > 6 & hp > 100)
head(mtcars_filtered_advanced)
```

```
##                      mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.44 17.02  0  0    3    2
## Duster 360          14.3   8 360.0 245 3.21 3.57 15.84  0  0    3    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.07 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.73 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.78 18.00  0  0    3    3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.25 17.98  0  0    3    4
```

Task 3:
Exercise: Filter cars that have either 4 or 8 cylinders.

```
# Your code here
```

Part 3: Creating New Variables with mutate()

3.1 Basic Mutate

The `mutate()` function creates or modifies columns. For example, to create a new column representing horsepower per weight:

```
mtcars_new_var <- mtcars %>%
  mutate(hp_per_wt = hp / wt)
head(mtcars_new_var)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb hp_per_wt
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4  41.98473
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4  38.26087
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1  40.08621
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1  34.21462
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2  50.87209
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1  30.34682
```

3.2 Creating Multiple New Columns

You can create multiple columns in one go. For example, add `hp_per_wt` and a scaled version of `mpg`:

```
mtcars_multi_mutate <- mtcars %>%
  mutate(hp_per_wt = hp / wt,
         scaled_mpg = scale(mpg))
head(mtcars_multi_mutate)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb hp_per_wt
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4  41.98473
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4  38.26087
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1  40.08621
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1  34.21462
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2  50.87209
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1  30.34682
##                   scaled_mpg
## Mazda RX4          0.1508848
## Mazda RX4 Wag      0.1508848
## Datsun 710         0.4495434
## Hornet 4 Drive     0.2172534
## Hornet Sportabout -0.2307345
## Valiant           -0.3302874
```

3.3 Conditional Mutate

Create new columns based on conditions using `if_else()`. For example, classify cars as “High HP” or “Low HP”:

```
mtcars_classified <- mtcars %>%
  mutate(hp_class = if_else(hp > 150, "High HP", "Low HP"))
head(mtcars_classified)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb hp_class
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4   Low HP
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4   Low HP
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1   Low HP
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1   Low HP
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2  High HP
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1   Low HP
```

Task 4:
Exercise: Create a new variable classifying cars as "Heavy" or "Light" based on their weight (wt).

```
# Your code here
```

3.4 Advanced: Using case_when()

Use `case_when()` for multiple conditions. For example, classify cars into weight categories:

```
mtcars_weight_class <- mtcars %>%
  mutate(weight_class = case_when(
    wt < 2.5 ~ "Light",
    wt >= 2.5 & wt < 3.5 ~ "Medium",
    wt >= 3.5 ~ "Heavy"
  ))
head(mtcars_weight_class)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
##                   weight_class
## Mazda RX4               Medium
## Mazda RX4 Wag           Medium
## Datsun 710               Light
## Hornet 4 Drive          Medium
## Hornet Sportabout       Medium
## Valiant                 Medium
```

3.5 Using mutate() with across()

Apply the same transformation to multiple columns using `across()`. For example, standardize the `mpg` and `hp` columns:

```
mtcars_scaled_vars <- mtcars %>%
  mutate(across(c(mpg, hp), scale))
head(mtcars_scaled_vars)
```

```
##                          mpg cyl disp         hp drat    wt  qsec vs am gear
## Mazda RX4          0.1508848   6  160 -0.5350928 3.90 2.620 16.46  0  1    4
## Mazda RX4 Wag      0.1508848   6  160 -0.5350928 3.90 2.875 17.02  0  1    4
## Datsun 710         0.4495434   4  108 -0.7830405 3.85 2.320 18.61  1  1    4
## Hornet 4 Drive     0.2172534   6  258 -0.5350928 3.08 3.215 19.44  1  0    3
## Hornet Sportabout -0.2307345   8  360  0.4129422 3.15 3.440 17.02  0  0    3
## Valiant           -0.3302874   6  225 -0.6080186 2.76 3.460 20.22  1  0    3
##                   carb
## Mazda RX4            4
## Mazda RX4 Wag        4
## Datsun 710           1
## Hornet 4 Drive       1
## Hornet Sportabout    2
## Valiant              1
```

Part 4: Sorting and Arranging Data

Use the `arrange()` function to sort your data. For example, to sort by `mpg` in ascending order:

```r
mtcars_sorted <- mtcars %>%
  arrange(mpg)
head(mtcars_sorted)
```

```
##                      mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Cadillac Fleetwood  10.4   8  472 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8  460 215 3.00 5.424 17.82  0  0    3    4
## Camaro Z28          13.3   8  350 245 3.73 3.840 15.41  0  0    3    4
## Duster 360          14.3   8  360 245 3.21 3.570 15.84  0  0    3    4
## Chrysler Imperial   14.7   8  440 230 3.23 5.345 17.42  0  0    3    4
## Maserati Bora       15.0   8  301 335 3.54 3.570 14.60  0  1    5    8
```

Task 5:
Exercise: Arrange the cars by horsepower (`hp`) in descending order.

```r
mtcars_sorted_desc <- mtcars %>%
  arrange(desc(hp))
head(mtcars_sorted_desc)
```

```
##                      mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Maserati Bora       15.0   8  301 335 3.54 3.570 14.60  0  1    5    8
## Ford Pantera L      15.8   8  351 264 4.22 3.170 14.50  0  1    5    4
## Duster 360          14.3   8  360 245 3.21 3.570 15.84  0  0    3    4
## Camaro Z28          13.3   8  350 245 3.73 3.840 15.41  0  0    3    4
## Chrysler Imperial   14.7   8  440 230 3.23 5.345 17.42  0  0    3    4
## Lincoln Continental 10.4   8  460 215 3.00 5.424 17.82  0  0    3    4
```

Part 5: Joining Data

Sometimes you'll need to combine data from two sources. dplyr offers functions like `left_join()`, `inner_join()`, etc. For example, suppose we have another dataset:

```r
# Create a simple data frame with car models and a new variable
car_info <- tibble(
  model = rownames(mtcars),
  origin = rep(c("USA", "Europe", "Japan"), length.out = nrow(mtcars))
)

# Join mtcars with car_info by converting row names to a column
mtcars_joined <- mtcars %>%
  rownames_to_column(var = "model") %>%
  left_join(car_info, by = "model")
head(mtcars_joined)
```

```
##               model  mpg cyl disp  hp drat    wt  qsec vs am gear carb origin
## 1         Mazda RX4 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4    USA
## 2     Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4 Europe
## 3        Datsun 710 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1  Japan
## 4    Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1    USA
## 5 Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2 Europe
## 6           Valiant 18.1   6  225 105 2.76 3.460 20.22  1  0    3    1  Japan
```

Tip: Use `left_join()` when you want to keep all observations from your main dataset.

Part 7: Exercises and Further Exploration

Now it's your turn! Try writing your own dplyr code: - Experiment with different filtering conditions. - Create new variables based on your own criteria. - Explore additional joins such as `right_join()` or `full_join()` with custom datasets.

```r
# Your exercise code here
```
