codecademy

Learn R: Aggregates

group_by() with mutate()

Combining `dplyr`'s `group_by()` and `mutate()` functions allows for the creation of new variables that involve per-group metrics in their calculation. Grouping a data frame by a selection of columns followed by a call to `mutate()` allows for the creation of new columns based on per-group summary functions. Given a `students` data frame, to add a new column containing the difference between a student's score and the mean `math_score` for each student's respective age group:

```
students %>%
  group_by(age) %>%
  mutate(diff_from_age_mean = math_
```

- `group_by()` groups the data frame by `age`
- `mutate()` will add a new column `diff_from_age_mean` which is calculated as the difference between a row's individual `math_score` and the `mean(math_score)` for that row's age-group

dplyr group_by()

dplyr 's `group_by()` function can group together rows of a data frame with the same value(s) in either a specified column or multiple columns, allowing for the application of summary functions on the individual groups.

`group_by()` changes the unit of analysis from a complete dataset to individual groups.

For example, consider a data frame `countries`. To find the mean and standard deviation of the `population` column grouped by `continent`:

```
countries %>%
  group_by(continent) %>%
  summarise(mean_pop = mean(popula
                        sd_pop = sd(
```

To find the mean `math_score` and `reading_score` by `age` and `gender` from a `students` data frame:

```
students %>%
  group_by(age, gender) %>%
  summarise(mean_math_score = mean
                          mean_reading_
```

filter() with group_by()

Combining dplyr 's group_by() and filter() functions allows for the filtering of rows of a data frame based on per-group metrics.

Grouping a data frame by a selection of columns followed by a call to filter() allows for filtering a data frame based on per-group summary functions.

Given a students data frame, to keep all the rows of students whose per-age average math_score is less than 80 :

```
students %>%
  group_by(age) %>%
  filter(mean(math_score, na.rm = T
```

- group_by() groups the data frame by age
- filter() will keep all the rows of the data frame whose per-group (per-age) average math_score is greater than 80

dplyr summarise()

dplyr 's `summarise()` function can collapse a data frame to a single row, summarising the specified columns by applying a summary function.
Summary functions, or functions that take a vector of values and return a single value, include: `mean()`, `median()`, `sd()` (standard deviation), `var()` (variance), `min()`, `max()`, `IQR()` (interquartile range), `n_distinct()` (number of unique values), and `sum()`.
For example, to find the mean `population` and `gdp` from `countries` data frame:

```
countries %>%
  summarise(mean_pop = mean(popula
```

To find the mean and standard deviation of both the `population` and `gdp` columns from a `countries` data frame:

```
countries %>%
  summarise(mean_pop = mean(popula
                        sd_pop = sd(
                        mean_gdp = me
                        sd_gdp = sd(
```

dplyr package

The dplyr package provides functions that perform data manipulation operations oriented to explore and manipulate datasets. At the most basic level, the package functions refers to data manipulation "verbs" such as select, filter, mutate, arrange, summarize among others that allow to chain multiple steps in a few lines of code. The dplyr package is suitable to work with a single dataset as well as to achieve complex results in large datasets.