Managing Data Frames with the dplyr package

x %in% y:

This will select every row where x is one of the values in y.

The dplyr Package

- The dplyr package was developed by Hadley Wickham of RStudio and is an optimized and distilled version of his plyr package.
- The dplyr package does not provide any "new" functionality to R per se, in the sense that everything dplyr does could already be done with base R, but it greatly simplifies existing functionality in R.
- Some of the key "verbs" provided by the dplyr package are:
  - **i.** select: return a subset of the columns of a data frame, using flexible notation.
  - **ii.** filter: extract a subset of rows from a data frame based on logical conditions.
  - **iii.** arrange: reorder rows of a data frame • rename: rename variables in a data frame.
  - **iv.** mutate: add new variables/columns or transform existing variables.
  - **v.** summarise / summarize: generate summary statistics of different variables in the data frame, possibly within strata.
  - **vi.** %>%: the "pipe" operator is used to connect multiple verb actions together into a pipeline.

Common dplyr Function Properties

All of the functions that we will discuss in this Chapter will have a few common characteristics. In particular,

1. The first argument is a data frame.
2. The subsequent arguments describe what to do with the data frame specified in the first argument, and you can refer to columns in the data frame directly without using the $ operator (just use the column names).
3. The return result of a function is a new data frame.

select()

- The select() function can be used to select columns of a data frame that you want to focus on.
- Often you'll have a large data frame containing "all" of the data, but any given analysis might only use a subset of variables or observations.
- The select() function allows you to get the few columns you might need.
- The select() function also allows a special syntax that allows you to specify variable names based on patterns. For example: ends_with(" "), starts_with(" ").

Filter rows with filter()

- The filter() function is used to extract subsets of rows from a data frame.
- It allows us to subset observations based on their values. The first argument is the name of the data frame. The second and subsequent arguments are the expressions that filter the data frame.
- This function is similar to the existing subset() function in R but is quite a bit faster.

Arrange rows with arrange()

- arrange works similarly to filter() except that instead of selecting rows, it changes their order.
- It takes a data frame and a set of column names (or more complicated expressions) to order by.
- If you provide more than one column name, each additional column will be used to break ties in the values of preceding columns.
- it is used to reorder rows of a data frame according to one of the variables/ columns.

rename()

- Renaming a variable in a data frame in R is surprisingly hard to do!
- The rename() function is designed to make this process easier.
- The syntax inside the rename() function is to have the new name on the left-hand side of the = sign and the old name on the right-hand side.

Add new variables with mutate()

- The mutate() function exists to compute transformations of variables in a data frame. Often,you want to create new variables that are derived from existing variables and mutate() provides a clean interface for doing that.

group_by()

- The group_by() function is used to generate summary statistics from the data frame within strata defined by a variable.
- The general operation here is a combination of splitting a data frame into separate pieces defined by a variable or group of variables (group_by()), and then applying a summary function across those subsets(summarize()).

Combining multiple operations with the pipeline operator %>%

- The pipeline operator %>% is very handy for stringing together multiple player functions in a sequence of operations.
-