

Learning Goals

- Use `group_by()` and `summarise()` to compute descriptive statistics for each category in a dataset.
- Use `n()`, `tally()`, and `count()` to count observations within groups and interpret those counts.
- Explain why `ungroup()` matters and predict how grouped vs. ungrouped operations change results.
- Group by multiple variables to create bivariate summaries and use `arrange()` / `filter()` to answer comparison questions.
- Use `n_distinct()`, the `slice()` family, ranking functions, and `lag()` / `lead()` for more advanced grouped analysis.

Key Definitions / Functions

- `group_by()`:

- `summarise()`:

- `n()`:

- `n_distinct()`:

- `slice()`:

- `rank()` and `min_rank()`:

- `lag()` and `lead()`:

Practice Problems

For each task below, write the R code you would use and briefly describe what you expect the output to look like.

1. Using the `iris` dataset, compute the number of observations and the mean `Sepal.Length` for each `Species`.

2. Using `iris`, create a grouped summary by `Species` and a new categorical variable `petal_group` defined as:

```
petal_group = "large" if Petal.Length >= 4.5, otherwise "small".
```

Report `n` and the mean `Petal.Width` for each combination, then arrange from largest to smallest mean `Petal.Width`.

3. Using `iris`, calculate a 5-number summary of `Sepal.Width` for each `Species` group.

4. Using `iris`, first sort the dataset by `Species` and then by `Sepal.Length`. Within each `Species`, create a new variable called `prev_sepal` using `lag(Sepal.Length)` and another variable called `diff_sepal = Sepal.Length - prev_sepal`. Then display the first 8 rows of the resulting data frame.