Merging Datasets Using R

Assil NOUREDDINE

2025-06-11

Abstract

This handout demonstrates how to merge two datasets using the `tidyverse` package in R. We use fictional employee and salary data to explore `left_join()` and `inner_join()`, visualize results, and summarize key findings. The project showcases essential skills for working with relational data and reporting results using Quarto.

Introduction

In data science, merging datasets is a common and critical operation. Often, information about a subject (such as an employee) is spread across multiple tables. Joining these datasets using common identifiers allows analysts to create comprehensive datasets for analysis.

In this project, we simulate this situation using two datasets:
- `employees.csv`: Contains basic employee information.
- `salaries.csv`: Contains salary data for some employees.

We'll merge these datasets and conduct simple analysis and visualization.

Loading Libraries and Data

```
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   3.5.2     v tibble    3.2.1
```

```
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- 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 becor
```

```r
# Load the datasets
employees <- read_csv("data/employees.csv")
```

```
Rows: 10 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): name, department
dbl (1): employee_id

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
salaries <- read_csv("data/salaries.csv")
```

```
Rows: 7 Columns: 2
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (2): employee_id, salary

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Structure of the Datasets

```r
glimpse(employees)
```

```
Rows: 10
Columns: 3
$ employee_id <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
$ name        <chr> "Alice", "Bob", "Charlie", "Diana", "Ethan", "Fiona", "Geo~
$ department  <chr> "HR", "IT", "IT", "Finance", "Marketing", "HR", "Finance",~
```

```r
glimpse(salaries)
```

```
Rows: 7
Columns: 2
$ employee_id <dbl> 1, 2, 4, 5, 6, 8, 9
$ salary      <dbl> 50000, 60000, 55000, 58000, 52000, 61000, 63000
```

Merging Datasets

We use left_join() to include all employees, even those who don't have a recorded salary, and inner_join() to include only employees who exist in both datasets.

Perform joins

```r
left_joined <- left_join(employees, salaries, by = "employee_id")
inner_joined <- inner_join(employees, salaries, by = "employee_id")
```

Summary Statistics by Department

We compute the average salary and the number of employees per department. Using left_join(), we ensure all employees are included in the analysis, even those with missing salary data.

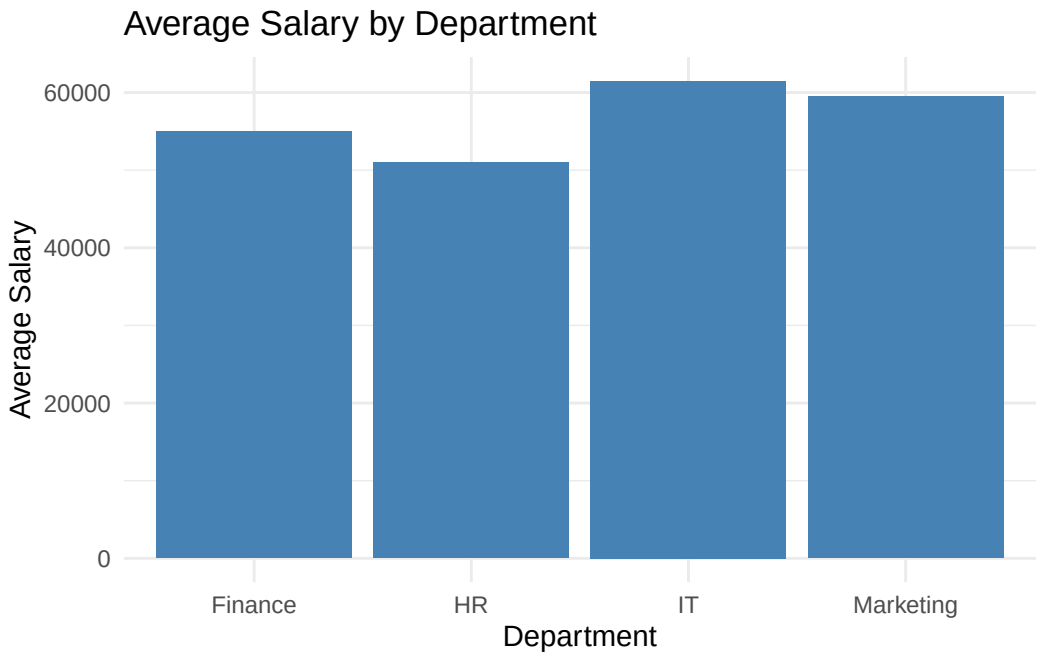
```r
left_joined %>%
  group_by(department) %>%
  summarise(
    avg_salary = mean(salary, na.rm = TRUE),
    count = n()
  )
```

```
# A tibble: 4 x 3
  department avg_salary count
  <chr>           <dbl> <int>
1 Finance         55000     2
2 HR              51000     3
3 IT              61500     3
4 Marketing       59500     2
```

Visualization: Average Salary by Department

The bar plot below shows the average salary per department.

```
left_joined %>%
  group_by(department) %>%
  summarise(avg_salary = mean(salary, na.rm = TRUE)) %>%
  ggplot(aes(x = department, y = avg_salary)) +
  geom_col(fill = "steelblue") +
  labs(
    title = "Average Salary by Department",
    x = "Department",
    y = "Average Salary"
  ) +
  theme_minimal()
```



Discussion

We observe differences in average salary across departments. This could reflect job roles, experience levels, or salary negotiation practices. The left_join() strategy helps include all employees in the analysis, even if some lack salary records.

Future Work Could:

Include more datasets (e.g., performance data) Explore mismatched joins Handle duplicate or missing IDs

Conclusion

This project showed how to combine datasets using dplyr joins in R. We practiced left_join() and inner_join(), and demonstrated data exploration and visualization techniques. These tools are powerful for preparing data in real-world projects.

References