Advanced Topics in R and RStudio

Yiheng Wu, Jeff Werst, Matt Steele

2025-09-15

Additional Resources

- R Markdown for RStudio
- R Markdown Cheatsheet
- R Markdown Reference Guide
- R Markdown Definitive Guide
- R Markdown Cookbook

Multi-Table Joins

What does it mean to "join tables"?

- Use a **key column** to join / combine two (or more) tables into one.

Why Joins?

- Data often lives in multiple tables (e.g., customer info in one, transactions in another).
- To analyze or model, we need to combine them into one dataframe.

What does key column means?

- A key column is a special column that exists in **both** tables and contains the same type of values
- It acts as a connector that tells the computer which rows in the two tables belong together and should be matched.
- Think of it like matching puzzle pieces: for each row in one table, we look for rows in another table with the same key value, and then put their details together into a single row.

Types of join

Table 1: Types of Joins

Left Join	Keep all rows from the left table, add matches from right
Right Join	Keep all rows from the right table, add matches from left
Inner Join	Keep only rows where the key exists in both tables
Full(Outer) Join	Keep all rows from both tables, even if unmatched

Tip: Choice of join affects your results — always check whether you might lose or duplicate rows.

Preparing Data

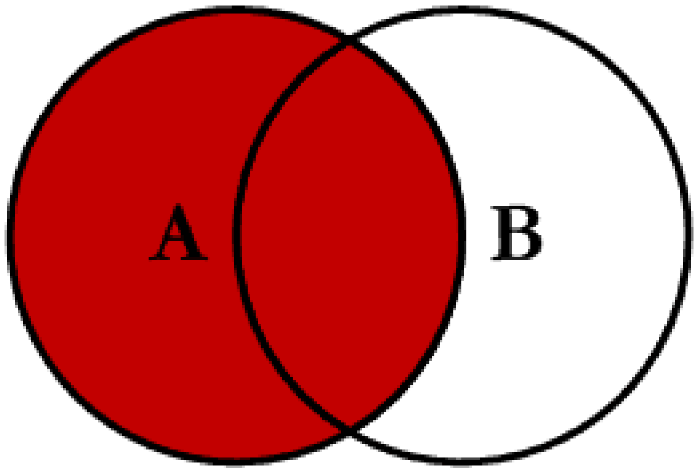
```r
library(dplyr)
library(tidyverse)
library(ggplot2)

students <- data.frame(
  student_id = c(1, 2, 3, 4, 5, 6, 7),
  name = c("Alice", "Bob", "Charlie", "David", "Eva", "Frank", "Grace"),
  department_id = c(201, 202, 201, 202, 202, 203, NA),
  gender = c("F", "M", "M", "M", "F", "M", "F")
)
```

```
scores <- data.frame(
  student_id = c(2, 3, 5, 6, 8),
  score = c(88, 92, 85, 63, 81)
)


departments <- data.frame(
  department_number = c(201, 202),
  department_name = c("Statistics", "Biology")
)
```

Left-Join

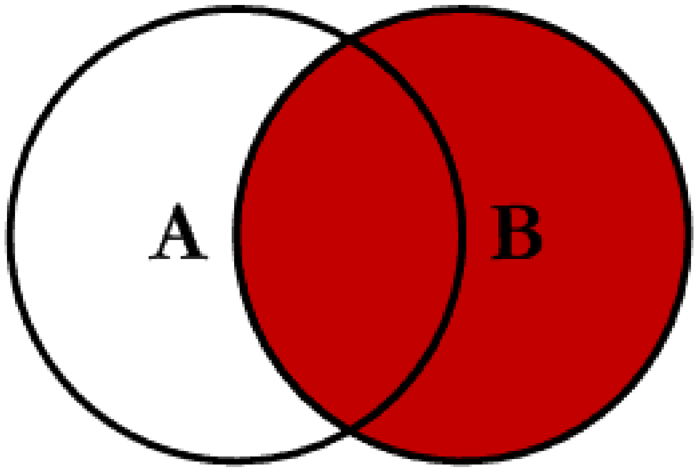


```
student_score <-students |>
  left_join(scores, by = 'student_id')

student_dept <- students |>
  left_join(departments, by=c('department_id'='department_number'))
```

```
student_score
student_dept
```

Right-Join

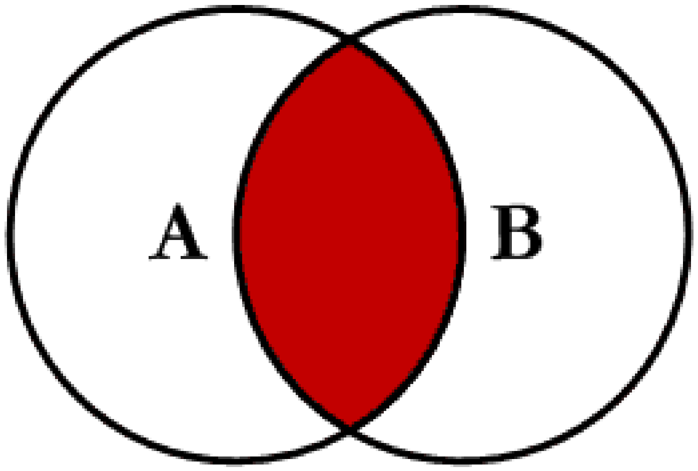


```
score_student <-scores |>
  left_join(students, by = 'student_id')

dept_student <- departments |>
  left_join(students, by=c('department_number'='department_id'))

score_student
dept_student
```

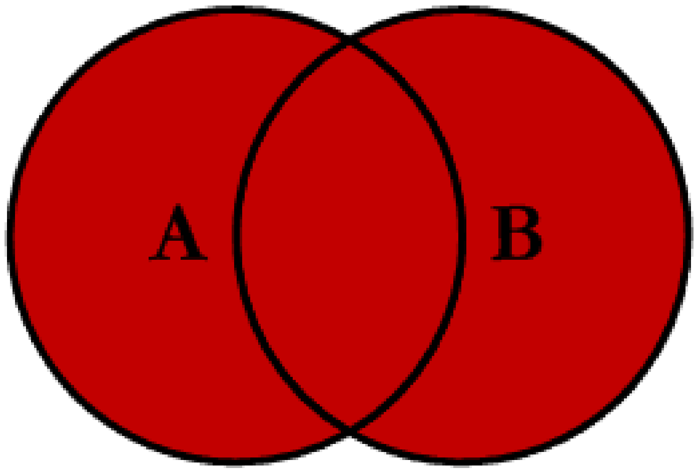
Inner-Join



```r
df_inner <- students |>
  inner_join(scores, by='student_id')

df_inner
```

Full(Outer)-Join



```r
df_full <- students |>
  full_join(scores,by='student_id')

df_full
```

Joining Multiple Tables

```r
df_all <- students |>
  full_join(scores,by='student_id') |>
  full_join(departments, by=c('department_id'='department_number'))

df_all
```

Data Wrangling

```r
df_all |>
  drop_na() |>
  group_by(gender) |>
  summarise(avg_score=mean(score))
```

About R Markdown

R Markdown allows you to blend formatted prose with code to create reproducible scientific documents that can be outputted in a HTML, PDF, and MS Word document.

Clicking on the **Knit** button in the editor toolbar will generate a document that includes both the content as well as the output of any embedded R code chunks within the document.

- Global Options
- Markdown Quick Reference (Help)

Why Bother?

1. Encourages you to document your analysis
2. Provides a non-proprietary format that you can easily store, preserve, document with metadata, and retrieve at later dates.
3. Reproducibility means that you can share the document with colleagues and peers to check errors or to collaborate easily. R Markdown even allows for multiple coding languages to be used in a single document.
4. Create reports/documents that are dynamically generated from you data and can be easily revised. R Markdown documents are dynamic and an errors or issues with the coding can be made with little work on the user's end.

- No longer do you need to re-code and re-paste

YAML Header

```
- YAML AIN'T MARKUP LANGUAGE
```

This is the metadata area for your document and it also determines how the document is rendered when you knit it. It's default fields are **title**, **author**, **date**, and **output**. But you can add more fields.

Available fields for YAML

About YAML:

- White spaces matter: indents indicate the contents are *child* of the level above
  - Spaces not tabs
- Boolean operators: true/false is lowercase
- true/false ~ yes/no
- Entries can include executable code
  - "2025-09-15"
- Most common outputs are *html_document*, *pdf_document*, and *word_document*
  - Full listing of available formats
  - For example, if you are interested in creating an interactive dashboard you would want to use the flexboard package output

```
# Help with HTML header options

?html_document
```

Formating Options

The following will provide ways for you to format your text/prose within the document that you are editing

```
#| label: formatting
#| eval: false


# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6



# Italics - *I am italic - mamma mia*

# Bold - **I am bold**

# Hyperlink - You can learn more about [RMarkdown here](https://rmarkdown.rstudio.com/)

# Image - ![Spongebob](spongebob.jpg)

# Footnotes - [^1]: This sentence is a footnote

# Block quote

# > "You miss 100% of the shots you do not take. - Wayne Gretsky" - Michael Scott

# Unordered lists:

# -   apple
# -   pear
# -   orange
# -   bear
  #  -   orange bear
  #  -   apple pear

# Ordered lists:

# 1.  Apple
# 2.  Pear
# 3.  Orange Bear
```

Document Editors

You can change the way that you edit the document by using the **Source** or **Visual** tab on the editor toolbar.

Source

- Allows you view the document in code view

Visual

- Allows you to view the document with markups
- Allows basic WSYWIG

Code Chunks

Code chunks allow you to include code from multiple languages into your narration.

You can insert a chunk code by:

- CTRL + ALT + I (PC)
- COMMAND + OPTIONS + I (MAC)
- Use **Add Chunk** command in editor toolbar

Let's add a code chunk that allows us to see the data set mtcars

Running a Code Chunk You can run a code chunk by:

- CTRL + SHIFT + ENTER (PC)
- COMMAND + SHIFT + ENTER (MAC)
- Run button in Code Chunk
- Run button in editor toolbar

Customize Chunk Code

Chunk Cog Wheel

- Allows you to rename the chunk so it can be easily located
- Allows you to set message and warning displays
- Allows you to adjust plot sizes

Let's rename our code chunk above

```
# I would encourage users to manually enter their labels. It is clearer for another user to view and cleaner for your presentatio
```

Manual Entry

Include Include allows you to include or not include the chunk code in the final product when knitted.

include =

Let's create a chunk code that sets our current working directory but does not display the code or output in our final product using include. Hint: Set the working directory with the command - setwd()

Eval Eval tells RStudio to either run or not run a code chunk when the document is knitted

eval =

Let's install the CRAN package Tidyverse. But since this is a one time operation, let's preface that this code is not run when the document is knitted.

Message Some commands, like loading a package, will display messages after the code is run. You can choose whether or not you want the message to be displayed in the knitted documents

message =

Let's load the tidyverse package because we will need functions in it to run future code in the report. However, let's set it so the load message does not appear when the document is knitted but the code is displayed so a person who we are collaborating with can see that we are using that package.

Echo Echo allows you to show the output of the code that has been run, but not to show the code chunk when the document is knitted

echo =

Let's get the results of a line of code without displaying the code in the report.

Inline Code

You can include coding within the body of your work using inline code using the backtick (') button on your keyboard

Let's include inline code with the mean of the mpg variable in the mtcars dataset as well as the number of observations of the variable.

The average miles per gallon from the cars dataset is 20.090625 based on 32 observations.

Plots

In addition to adding code and outputs of the code, you can also set up data visualization to be displayed in your documents.

Here we will add a histogram of the dataset for the variable mpg. And we will use R Markdown to determine the size of the figure as well as give it a captions. Additionally, as we have learned already, we will use echo=FALSE to display only the output and not the code.

```
#| label:
#| echo: false
#| message: false
#| fig.align: 'center'
#| fig.width: 10
```

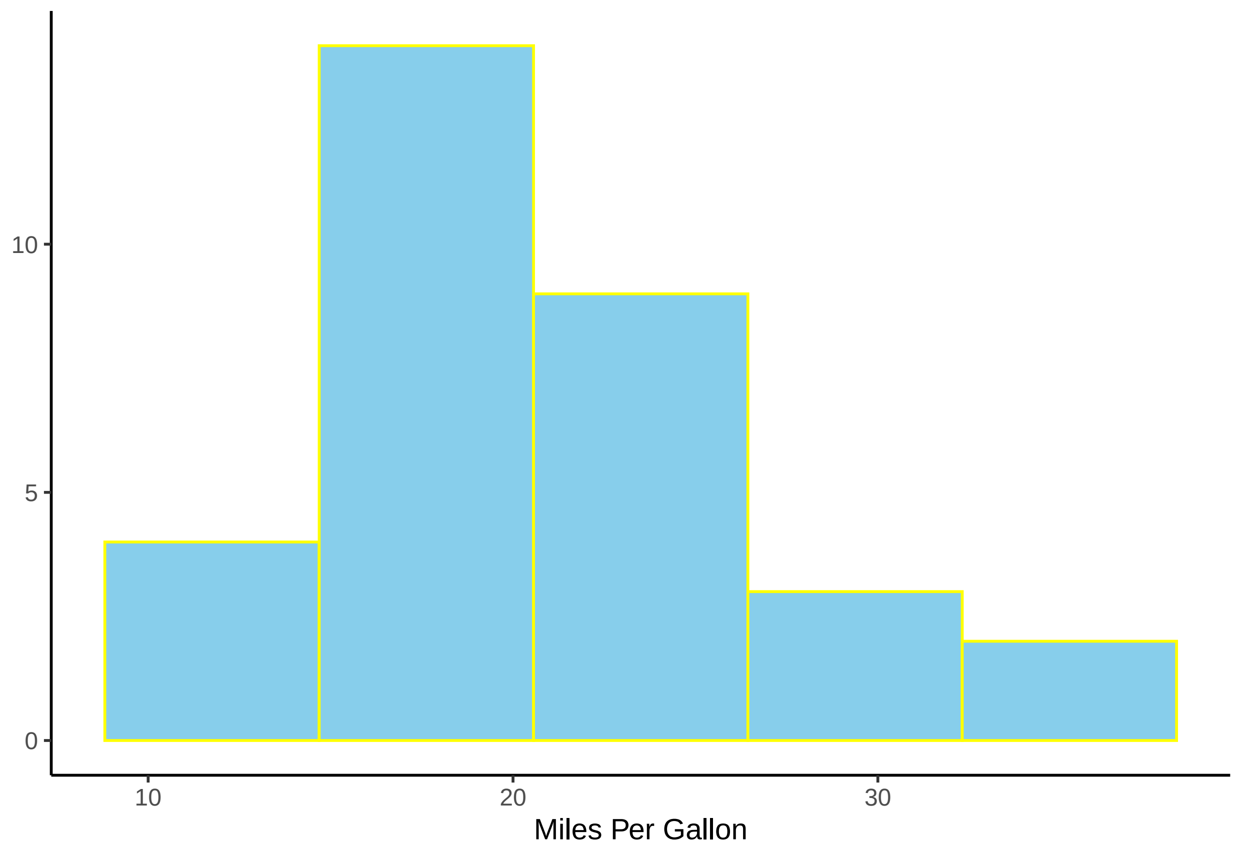
```r
#| fig.cap: "Figure 6.2: MPG Distribution"

library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.5.1
```

```
## -- 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    3.3.0
## 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 become errors
```

```r
mtcars.hist <- ggplot(mtcars, aes(x=mpg))
mtcars.hist +
  geom_histogram(bins = 5, color = "yellow", fill = "skyblue") +
    labs(x = "Miles Per Gallon",
       y = NULL) +
  theme_classic()
```



Citations

R Markdown allows you to insert citations as well as work with citation managers such as Zotero and CiteDrive. Once a citation is added to the document, it will automatically populate in a bibliography at the end of the document.

Insert Citations into your document:

- Visual Mode: Insert > Citation
- Source Mode: [@auerbach2021] or (See [@grolemund])
- Visual Mode: @ will show you available citations

When a Citation is generated:

- A new .bib file will be created in the current working directory and will be attached to the document in the YAML header
- The default format for the citations is **Chicago Turabian**. If you want to change the format you will need to download the proper .csl file and add it to your working directory and add a csl field to your YAML header
  - Zotero Library
  - Citation Visual Editor

Let's add APA 7th Ed. Citation Format to our Working Directory and YAML header

Let's try and find and enter the citation for the following article

- 10.1016/j.jvs.2021.03.055

References