PLS 120: Applied Statistics in Agricultural Sciences

Data Manipulation with dplyr

Week 3 Tutorial Guide

Mohammadreza Narimani
Department of Biological and Agricultural Engineering
University of California, Davis

mnarimani@ucdavis.edu

October 2025

Contents

Important Links

Essential Course Resources

Course Website

All course materials available at:

Course Website Link

Interactive Binder Environment

Access Week 3 lab materials:

Week 3 Binder Link

Welcome to Week 3: Data Manipulation with dplyr

This week, we explore **data manipulation using dplyr** and the **tidyverse ecosystem** - essential tools for organizing, cleaning, and visualizing agricultural data. You'll learn to filter, select, arrange, and transform data efficiently!

Key Data Manipulation Concepts

Basic Data Subsetting

Understanding how to extract specific parts of your data is fundamental to data analysis.

3.1.1 Bracket Notation

Syntax: `data[row, column]`

Examples:
`data[1,1]` - Single cell
`data[1:5, ]` - First 5 rows
`data[, 1:3]` - First 3 columns
`data[1:10, 2:4]` - Specific rows and columns

The Power of Pipes (%>%)

Concept: Chain operations together for readable code

Think of pipes as "then": Take data, **then** filter, **then** select

Example: `data %>% filter(Species == "setosa") %>% select(Sepal.Length)`

Essential dplyr Functions

Filtering Data

4.1.1 filter() Function

Purpose: Subset rows based on conditions

Syntax: `filter(data, condition)`

Examples:
`filter(Species == "setosa")` - Exact match
`filter(Sepal.Length > 5)` - Numerical condition
`filter(Species %in% c("setosa", "virginica"))` - Multiple values

4.1.2 slice() Functions

Purpose: Select rows by position

Functions:
`slice(10:20)` - Rows 10 to 20
`slice_head(n=5)` - First 5 rows
`slice_tail(n=5)` - Last 5 rows
`slice_sample(n=10)` - Random 10 rows

Selecting Columns

4.2.1 select() Function

Purpose: Choose specific columns

Basic Selection:
`select(Sepal.Length, Species)` - By name
`select(1, 5)` - By position
`select(1:3)` - Range of columns

4.2.2 Advanced Selection Helpers

Pattern Matching:
`starts_with("Sepal")` - Columns starting with "Sepal"
`ends_with("Length")` - Columns ending with "Length"
`contains("Petal")` - Columns containing "Petal"
`matches(".*Width")` - Regular expression matching

Data Transformation

4.3.1 arrange() Function

Purpose: Sort data by variables

Examples:
`arrange(Sepal.Length)` - Ascending order
`arrange(desc(Sepal.Length))` - Descending order
`arrange(Species, Sepal.Length)` - Multiple variables

4.3.2 mutate() Function

Purpose: Create new variables or transform existing ones

Examples:

```
mutate(Sepal.Area = Sepal.Length * Sepal.Width)
mutate(Length_mm = Sepal.Length * 10)
mutate(Size_Category = ifelse(Sepal.Length > 5, "Large", "Small"))
```

4.3.3 rename() Function

Purpose: Change column names

Syntax: `rename(new_name = old_name)`

Example: `rename(Sepal_Length = Sepal.Length, Plant_Species = Species)`

Grouping and Summarizing

4.4.1 group_by() and summarize()

Purpose: Calculate statistics for different groups

Workflow:
1. Group data by categorical variable
2. Calculate summary statistics for each group

Example:

```
data %>% group_by(Species) %>%
summarize(mean_length = mean(Sepal.Length))
```

Data Visualization

ggplot2 Fundamentals

5.1.1 Basic ggplot Structure

Components:
1. `ggplot(data, aes(x = variable, y = variable))` - Base layer
2. `+ geom_*()` - Add geometric objects
3. `+ labs()` - Add labels and titles

Example:
```
ggplot(data, aes(x = Species, y = Sepal.Length)) + geom_boxplot()
```

5.1.2 Common Plot Types

Histograms: `geom_histogram()` - Distribution of single variable
Boxplots: `geom_boxplot()` - Compare groups
Scatter plots: `geom_point()` - Relationship between variables
Bar plots: `geom_bar()` - Count categorical data
Density plots: `geom_density()` - Smooth distribution curves

Base R Plotting

5.2.1 Quick Visualization Functions

Histograms: `hist(data$variable)`
Boxplots: `boxplot(variable ~ group, data = data)`
Scatter plots: `plot(x, y)`
Stem-and-leaf: `stem(data$variable)`

Data Cleaning Techniques

Handling Non-Numeric Data

6.1.1 String Manipulation

Remove non-numeric characters:
```
str_replace_all(text, "[^0-9]", "")
```

Convert to numeric:
```
as.integer(character_vector)
as.numeric(character_vector)
```

6.1.2 Missing Value Treatment

Remove missing values: `na.omit(data)`
Check for missing values: `is.na(data)`
Count missing values: `sum(is.na(data))`

Assignment 3 Overview

Assignment Structure (20 points total)

1. **Part 1: LA Crime Data Analysis (6 points)**
   - Load and filter data by gender
   - Create comparative boxplots
   - Interpret statistical differences
2. **Part 2: SAT Dataset Processing (9 points)**
   - Import and inspect data
   - Create random subsets
   - Clean non-numeric values
   - Extract specific columns
3. **Part 3: Distribution Analysis (5 points)**
   - Create stem-and-leaf plots
   - Analyze distribution patterns
   - Identify outliers and central tendency

Agricultural Applications

Real-World Applications:

- **Crop Yield Analysis** - Filter by variety, location, season
- **Soil Sample Processing** - Clean mixed numeric/text data
- **Weather Pattern Analysis** - Group by month, calculate averages
- **Livestock Performance** - Compare treatments, identify outliers
- **Quality Control** - Monitor product consistency over time
- **Field Trial Analysis** - Subset by treatment groups

Getting Started

1. Launch Week 3 Binder environment
2. Navigate to `class_activity` folder
3. Open `Week3_Data_Manipulation.ipynb`
4. Work through interactive exercises

Learning Objectives

By the end of this week, you will be able to:

- Master data subsetting with brackets and logical conditions
- Apply dplyr functions for efficient data manipulation
- Chain operations using pipes for readable workflows
- Select columns using helper functions
- Clean real-world data with mixed data types
- Create visualizations to understand data patterns
- Interpret statistical distributions and identify outliers

Need Help?

Mohammadreza Narimani
Email: mnarimani@ucdavis.edu
Department of Biological and Agricultural Engineering, UC Davis
Office Hours: Thursdays 10 AM - 12 PM (Zoom)