Table of contents

Data Wrangling with SQL

Student Edition

By Kimberly Seefeld, EdD

Published under **DataJoyAI** imprint

Companion resources for this book are available at:

https://datajoyai.com

Edition: First Edition
ISBN: 978-1-969233-34-0

1 Introduction to Data Wrangling

1.1 Learning Outcomes

1-1: Define data wrangling and explain its critical role in preparing datasets for analysis

1-2: Identify the key stages of the data lifecycle and understand how data preparation impacts analysis quality

1-3: Compare the strengths of different data wrangling tools including Excel, SQL, R, and Python

1-4: Recognize common data quality challenges and understand systematic approaches to address them

1-5: Develop a framework for choosing appropriate tools based on data characteristics and project requirements

1.2 Introduction

Picture this: You're a marketing analyst at a growing e-commerce company, and your boss asks you to analyze customer purchasing patterns to improve the company's email marketing campaigns. You're excited—this is exactly the kind of project that could showcase your analytical skills and drive real business impact. The IT department provides you with three months of customer data, and you eagerly dive in.

Your excitement quickly turns to frustration. Customer names appear in dozens of different formats—some are "John Smith," others are "Smith, John," and still others are "J. Smith" or even "JOHN SMITH." Purchase dates range from perfectly reasonable to completely impossible, like orders placed on February 30th. Some customers have complete profiles with addresses, phone numbers, and purchase history, while others exist as little more than an email address. The product data is equally messy, with the same item appearing under multiple names and categories.

What seemed like a straightforward analysis project has suddenly become a complex puzzle of data detective work. You realize that before you can answer any business questions, you need to solve a more fundamental problem: transforming this messy, inconsistent data into something you can actually analyze.

Welcome to the world of data wrangling—the often unglamorous but absolutely essential process of cleaning, transforming, and preparing data for analysis. While it may not be the most exciting part of data analysis, it's arguably the most important. Every insight you generate, every recommendation you make, and every decision that flows from your analysis depends entirely on the quality of your data preparation.

1.3 The Reality of Real-World Data

1.3.1 Why Data Gets Messy

Data doesn't start life in perfect condition. In fact, the opposite is almost always true. Real-world data becomes messy through a combination of human error, system limitations, and the natural complexity of capturing information about the world around us.

Consider how customer information enters a typical business system. Sales representatives might enter names differently based on their personal preferences or training. Online forms might allow customers to enter information in various formats. Different systems might store the same information using different field names or data types. When a company grows through acquisitions, it inherits data from systems that were never designed to work together.

The problem compounds over time. A small inconsistency in how dates are formatted might seem trivial when you have a hundred records, but it becomes a major headache when you're dealing with hundreds of thousands of transactions. A missing field that affects only a few customers initially might represent a significant gap in your analysis when your customer base grows.

Understanding why data gets messy helps you approach data wrangling with the right mindset. You're not dealing with careless mistakes that could have been easily avoided—you're dealing with the inevitable result of complex systems, human behavior, and changing business requirements over time.

1.3.2 The Cost of Poor Data Quality

Poor data quality isn't just an inconvenience—it has real business consequences. Studies consistently show that organizations lose significant revenue due to data quality issues, with some estimates suggesting that poor data quality costs the average organization millions of dollars annually.

The costs manifest in multiple ways. Incorrect customer information leads to failed deliveries and frustrated customers. Inconsistent product data makes it difficult to track inventory accurately or analyze sales trends. Missing or inaccurate financial data can lead to poor business decisions or regulatory compliance issues.

Perhaps more importantly, poor data quality undermines confidence in analysis and decision-making. When business leaders can't trust the data, they're less likely to rely on data-driven insights, falling back on intuition and experience instead. This creates a vicious cycle where poor data quality leads to less data usage, which reduces investment in data quality improvement.

1.4 What Data Wrangling Really Means

1.4.1 Beyond Simple Cleaning

Data wrangling encompasses much more than just fixing obvious errors or filling in missing values. It's a comprehensive process that involves understanding your data's structure and limitations, making informed decisions about how to handle quality issues, and transforming data into formats that enable effective analysis.

Think of data wrangling as similar to preparing ingredients for a complex meal. A skilled chef doesn't just wash vegetables and hope for the best. They carefully select the best ingredients, understand how different preparation methods affect flavor and texture, and organize everything to create a cohesive final dish. Similarly, effective data wrangling requires understanding your data's characteristics, knowing how different transformation choices affect your analysis, and organizing information to support your analytical goals.

The "wrangling" metaphor captures the sometimes unpredictable and challenging nature of working with real-world data. Just as wrangling livestock requires patience, skill, and adaptability, working with messy data requires persistence, creativity, and the ability to adjust your approach as you discover new challenges.

1.4.2 The Art and Science of Data Preparation

Data wrangling combines technical skills with business judgment. The technical aspects involve knowing how to use tools effectively, understanding data formats and structures, and implementing transformations efficiently. The judgment aspects involve deciding how to handle ambiguous cases, determining what level of data quality is sufficient for your purposes, and balancing the time invested in data preparation against the value of improved data quality.

Consider the challenge of handling duplicate customer records. The technical aspect involves writing code or using tools to identify potential duplicates based on matching names, addresses, or other identifiers. The judgment aspect involves deciding what constitutes a "match"—should "John Smith" and "Jon Smith" be considered the same person? What about "John Smith" at "123 Main St" and "J. Smith" at "123 Main Street"?

These decisions require understanding your business context, the intended use of the data, and the trade-offs between different approaches. There's rarely a single "correct" way to handle data quality issues—instead, there are different approaches with different advantages and limitations.

1.5 The Data Wrangling Process

1.5.1 Discovery and Assessment

Every data wrangling project begins with discovery—understanding what data you have, where it came from, and what condition it's in. This exploration phase is crucial because it informs all subsequent decisions about how to clean and transform your data.

Discovery involves examining data structure, identifying patterns in missing values, understanding the range and distribution of different variables, and assessing the overall quality of your dataset. You might discover that certain fields are consistently missing for specific customer segments, that date fields contain impossible values, or that categorical variables have more unique values than expected.

This assessment phase often reveals surprises. Data that appeared clean in small samples might show significant quality issues when examined comprehensively. Fields that seemed straightforward might contain unexpected complexity. Understanding these issues early helps you plan an effective data preparation strategy.

1.5.2 Cleaning and Standardization

Once you understand your data's characteristics and limitations, you can begin the systematic process of cleaning and standardization. This involves correcting obvious errors, standardizing formats, and handling missing values in ways that support your analytical objectives.

Cleaning decisions should be guided by your understanding of the data's intended use. If you're preparing data for a customer segmentation analysis, ensuring consistent customer identifiers might be more important than having complete address information. If you're analyzing sales trends over time, having accurate and consistent date information becomes crucial.

Standardization involves establishing consistent formats and conventions across your dataset. This might mean converting all names to a standard case format, ensuring dates follow a consistent pattern, or establishing standard categories for products or services. The goal is creating consistency that enables effective analysis while preserving the meaningful information in your data.

1.5.3 Transformation and Enhancement

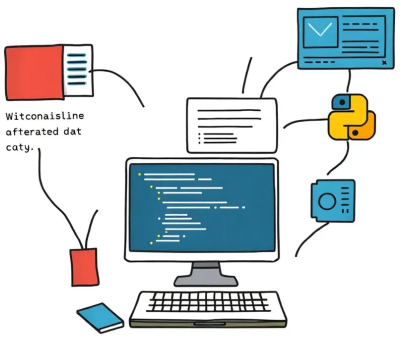
Beyond basic cleaning, data wrangling often involves transforming data to better support your analytical objectives. This might include creating new variables

that combine information from multiple fields, aggregating data at different levels of detail, or restructuring data to match the requirements of your analysis tools.

Transformation decisions should be driven by your analytical goals rather than technical convenience. If your analysis requires understanding customer behavior over time, you might need to transform transaction-level data into customer-level summaries. If you're comparing performance across different regions, you might need to standardize geographic information and create regional groupings.

Enhancement involves adding value to your data through external sources or derived calculations. This might include adding demographic information based on zip codes, calculating customer lifetime value from transaction history, or creating seasonal indicators from date information.

1.6 The Multi-Tool Reality



1.6.1 Why No Single Tool Rules

Modern data wrangling rarely involves using just one tool from start to finish. Instead, effective data preparation typically combines multiple tools, each opti-

mized for different aspects of the process. Understanding when and how to use different tools is a key skill for effective data wrangling.

Different tools excel at different tasks. Spreadsheet software provides intuitive interfaces for data exploration and manual cleaning tasks. Database query languages like SQL offer powerful capabilities for working with large, structured datasets. Programming languages like R and Python provide flexibility for complex transformations and automated processing.

The choice of tools often depends on factors like data size, complexity, team skills, and integration requirements. A project involving a few thousand records might be handled entirely in Excel, while a project involving millions of records might require SQL for initial processing and Python for complex transformations.

1.6.2 Excel: The Universal Explorer

Despite the emergence of sophisticated programming tools, Excel remains a crucial component of most data wrangling workflows. Its visual interface makes it ideal for initial data exploration, quality assessment, and manual cleaning tasks that require human judgment.

Excel excels at helping you understand your data quickly. You can sort columns to see value ranges, use conditional formatting to highlight unusual values, and create simple charts to visualize data distributions. For many data wrangling tasks, this visual approach is more efficient than writing code.

Excel's limitations become apparent as data size and complexity increase. It struggles with datasets larger than a million rows and lacks the automation capabilities needed for complex, repeatable data preparation processes. However, even in projects that primarily use other tools, Excel often plays a valuable role in exploration and validation.

3.12 Lab 3-2 AI-Assisted Data Imputation

Missing data is an unavoidable reality in analytics. Customers skip questions, systems fail to record entries, and information is lost or corrupted. How you handle these gaps directly affects the quality of your analysis. Simple approaches—such as filling in missing values with averages or deleting incomplete rows—are easy to apply, but they often distort patterns and introduce bias.

AI-assisted imputation offers a more powerful alternative. Instead of relying on fixed rules, AI-based methods learn relationships within the data and use those learned patterns to estimate missing values. For example, if a customer's age is missing, an AI model can consider income, location, and spending behavior to infer a likely age. Because these methods incorporate multiple variables simultaneously, they often produce more realistic and context-aware estimates than basic techniques.

This lab walks through the full spectrum of imputation strategies in R—from simple replacements to statistical modeling, machine-learning-based imputation, and finally AI-assisted reasoning using external large language models. The goal is not only to show how each method works, but also to illustrate how imputation evolves from manual assumptions to fully automated pattern recognition.

3.12.1 Demonstration

3.12.1.1 Step 1 Creating a Dataset with Missing Values

We begin with a small dataset containing missing ages:

```
customer_data <- data.frame(
  name = c("John Smith", "Jane Doe", "Bob Wilson", "Alice Brown", "Mike Joh
  income = c(45000, 75000, 52000, 68000, 59000),
  age = c(NA, NA, 35, 42, NA)
)
```

```
customer_data
```

```
          name income age
1   John Smith  45000  NA
2     Jane Doe  75000  NA
3   Bob Wilson  52000  35
4  Alice Brown  68000  42
5 Mike Johnson  59000  NA
```

This simple dataset allows us to explore how different imputation strategies behave.

3.12.1.2 Step 2 Simple Imputation Using the Mean

The most basic approach is to replace missing values with the mean of the observed values:

```
mean_age <- mean(customer_data$age, na.rm = TRUE)

mean_age
```

```
[1] 38.5
```

```
customer_data$age_mean_imputed <- ifelse(
  is.na(customer_data$age),
  mean_age,
  customer_data$age
)

customer_data
```

```
          name income age age_mean_imputed
1   John Smith  45000  NA             38.5
```

```
2     Jane Doe  75000  NA            38.5
3   Bob Wilson  52000  35            35.0
4  Alice Brown  68000  42            42.0
5 Mike Johnson  59000  NA            38.5
```

This method is fast and easy, but it ignores relationships between variables and reduces natural variability.

3.12.1.3 Step 3: Conditional Imputation

Step 2 introduces conditional imputation, which is a more informed approach than simply replacing missing values with a single overall statistic like the mean. Instead of treating all missing ages the same, this method uses another variable—in this case, income—to guide the estimation process. The underlying assumption is that income and age are often related in real-world data, where higher income may be associated with older individuals due to increased work experience, while lower income may be more common among younger individuals who are earlier in their careers. Based on this relationship, missing age values are estimated using a simple rule: individuals with lower incomes are assigned younger estimated ages, while individuals with higher incomes are assigned older estimated ages. This approach adds context to the imputation process by incorporating domain-based reasoning rather than relying on a single global average.

```
customer_data$age_conditional <- ifelse(
  is.na(customer_data$age) & customer_data$income < 60000, 30,
  ifelse(
    is.na(customer_data$age) & customer_data$income >= 60000, 45,
    customer_data$age
  )
)

customer_data
```