# Common Data Errors and How to Fix Them

A practical guide with common data errors, how to detect them, and practical solutions — designed to support reliable, high-quality analysis using Excel, SQL, and Python.

Created by Astrid Villalobos
Economist | Data & Project Professional
Montreal, 2025

# How to Use This Guide

This guide is designed as a quick reference for data analysts and professionals who work with datasets across tools like Excel, SQL, and Python. Each section follows a simple and actionable structure:

- - Common Problem: A brief explanation of a typical data issue you might encounter (e.g., duplicate rows, inconsistent date formats, missing values).
- - How to Detect: Practical methods or commands to help you identify the issue, using examples from Excel formulas, SQL queries, or Python code.
- - How to Fix:  Suggested solutions or best practices to clean, transform, or standardize the data effectively.

You can read this guide from beginning to end or jump directly to the type of error you're currently working on. Whether you're validating datasets, preparing inputs for a model, or cleaning data for reporting, this document offers practical help you can apply immediately.

### Introduction

Working with data is rarely perfect. Data analysts constantly face errors and inconsistencies that can severely affect the accuracy and reliability of insights. Detecting and correcting these issues early is crucial to produce trustworthy results and informed decisions.

## 1. Date Format Errors

**Common problems:**

- Mixed formats: DD/MM/YYYY vs MM/DD/YYYY causing misinterpretation
- Dates stored as text preventing calculations or sorting
- Missing, invalid, or logically impossible dates (e.g., 31/02/2023)

**How to detect:**

- Identify text-formatted dates in Excel by filtering or ISDATE function
- In SQL, test casting strings to date type with TRY_CAST() or ISDATE()
- Use pd.to_datetime() with error coercion in Python to flag invalid dates

**How to fix:**

- Standardize dates to ISO format YYYY-MM-DD using Excel's Power Query or formula tools
- Convert text dates to proper date types in SQL/Python with error handling
- Manually review or impute missing or invalid dates based on business context
- pandas Documentation: to_datetime
-

## 2. Duplicate Records

**Common problems:**

- Exact duplicates, which inflate counts or cause bias
- Partial duplicates where unique identifiers are missing or inconsistent
- Entity duplicates due to variations in naming or ID

**How to detect:**

- Excel's "Remove Duplicates" and Conditional Formatting for quick checks
- SQL queries using GROUP BY and HAVING COUNT(*) > 1
- Python's df.duplicated(subset=[...]) to find duplicates on key columns

**How to fix:**

- Remove or flag duplicates after reviewing which record is most complete or recent
- Merge partial duplicates with data consolidation logic or fuzzy matching
- Establish unique identifiers and enforce them to prevent future duplicates and define primary keys or unique constraints where applicable.

## 3. Missing Values

**Common problems:**

- Blanks or NULLs in key columns impacting calculations
- Systematic missingness (e.g., data not collected for some periods or groups)
- Inconsistent representations (empty strings, zeros, or special codes like -999)

**How to detect:**

- Excel filters or formulas like COUNTA() and COUNTBLANK()
- SQL queries with WHERE column IS NULL or IS NOT NULL conditions
- Python's df.isnull().sum() and visualizations like heatmaps to detect missingness patterns

**How to fix:**

- Impute using statistical methods: mean, median, mode, or domain-specific values
- Use forward-fill or backward-fill for time series data
- Drop rows/columns if missing data is minimal and random
- Clearly document any imputation or data removal

-

### 4. Outliers and Extreme Values

**Common problems:**

- Data points that fall far outside expected or logical ranges
- Typographical errors (extra zeros, misplaced decimals)
- Measurement or entry errors causing skewed distributions

**How to detect:**

- Visualize data with boxplots, histograms, or scatter plots
- Calculate Z-scores (standard deviations from mean) or use the Interquartile Range (IQR) method
- SQL queries to filter values outside expected thresholds

**How to fix:**

- Verify outliers against source data or business knowledge
- Remove or cap extreme values (Winsorize) if justified
- Use robust statistical methods that reduce outlier influence

### 5. Inconsistent Categorical Data

**Common problems:**

- Variations in spelling, capitalization, or language in labels
- Use of multiple synonyms or abbreviations for the same category
- Mixed categories due to lack of standardization

**How to detect:**

- Use pivot tables or group counts in Excel to see variations
- SQL GROUP BY and COUNT(*) to find category variants
- Python’s .unique() or .value_counts() to identify inconsistencies

**How to fix:**

- Create a mapping dictionary or lookup table for standardization
- Use Excel’s Find & Replace or Data Validation lists
- Use Python’s .replace() or .map() to harmonize categories

- Consider maintaining a data dictionary to track standardized labels and prevent drift over time.

## 6. Encoding Issues:

**Common Problems:**

Special characters (e.g., é, ñ) appearing as gibberish (e.g., �) or import failures due to mismatched encodings (e.g., UTF-8 vs. ANSI).

**How to Detect:**

- Excel: Check for garbled text in imported CSVs or use Text Import Wizard.
- SQL: Query for non-standard characters with LIKE '%[^a-zA-Z0-9]%'.
- Python: Use chardet.detect() or try pd.read_csv(..., encoding='utf-8') and catch errors.

**How to Fix:**

- Re-save files with UTF-8 encoding in Excel or text editors.
- SQL: Use CONVERT() or specify CODEPAGE in BULK INSERT.
- Python: Read files with correct encoding (e.g., pd.read_csv(..., encoding='utf-8')) or use ftfy.fix_text()

## Tools Summary

| Task | Excel | SQL | Python (pandas) |
|---|---|---|---|
| Detect duplicates | Remove Duplicates, Conditional Formatting | GROUP BY + HAVING COUNT(*) > 1 | df.duplicated() |
| Fix date formats | DATEVALUE, Power Query | TRY_CAST, CONVERT | pd.to_datetime() |
| Handle missing data | Filter blanks, COUNTA | IS NULL, COALESCE | df.isnull(), fillna() |
| Identify outliers | Boxplots, Filters | WHERE clauses | Z-score, IQR |
| Standardize categories | Find & Replace, Validation lists | CASE statements | .replace(), .map() |
| Fix encoding issues | Text Import Wizard, Save as UTF-8 | CONVERT(), CODEPAGE in imports | pd.read_csv(..., encoding='utf-8'), ftfy |

### Final Tips

- - Always keep a backup of raw data before cleaning.
- - Document every cleaning step clearly for auditability and reproducibility.
- - Validate major corrections with subject matter experts or data owners.
- - Whenever possible, automate repetitive cleaning steps using scripts or tools.
- - Understand the business context behind data issues to choose appropriate fixes.
- - Always check file encoding when importing data from external sources to avoid character corruption.

### APA References

- - IBM. (n.d.). Handling missing data in analytics. IBM Documentation. Retrieved August 3, 2025, from https://www.ibm.com/docs/en/spss-statistics/31.0.0?topic=data-handling-missing
- - Microsoft. (n.d.). Date and time data types and functions (Transact-SQL). Microsoft Learn. Retrieved August 3, 2025, from https://learn.microsoft.com/en-us/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql?view=sql-server-ver17
- - National Institute of Standards and Technology. (n.d.). Outlier detection. Engineering Statistics Handbook. Retrieved August 3, 2025, from https://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm
- - PyPI. (n.d.). chardet: Character encoding detection. Retrieved August 3, 2025, from https://pypi.org/project/chardet/
- - The pandas development team. (n.d.-a). Categorical data. pandas Documentation. Retrieved August 3, 2025, from https://pandas.pydata.org/pandas-docs/stable/user_guide/categorical.html
- - The pandas development team. (n.d.-b). pandas.DataFrame.duplicated. pandas Documentation. Retrieved August 3, 2025, from https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html
- - The pandas development team. (n.d.-c). Working with missing data. pandas Documentation. Retrieved August 3, 2025, from https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html