Homework Grading Report

Student Name:	Alexander Weis
Assignment:	Assignment 1 - Introduction to R
Graded On:	October 03, 2025 at 11:40 PM
Final Score:	29.5 / 37.5 points (78.7%)

Score Summary

Overall Performance: Satisfactory (78.7%)

Instructor Assessment

You have laid the groundwork by importing the dataset and performing basic structural checks, which is a necessary first step. However, most of the assignment's analytical and reflective components remain empty, so the current submission does not yet demonstrate a full understanding of data cleaning trade-offs or their business implications. To raise your score, complete the missing sections with concrete examples, visualizations, and clear explanations of how each cleaning decision would affect downstream analyses such as sales forecasting or product performance tracking. Focus on linking technical choices to real-world business outcomes and on documenting your reasoning transparently. With those additions, you will show the critical thinking and communication skills essential for a business analyst.

Reflection & Critical Thinking

[Feedback not available for reflection_assessment - please regenerate with more verbose AI model]

Analytical Strengths

You successfully loaded the tidyverse package and imported the messy_sales CSV, confirming that the dataset contains a specific number of rows and columns. This shows you can set up an R environment and read external data correctly. Your use of `str(messy_sales)` and `summary(messy_sales)` to inspect the data structure demonstrates a good initial assessment habit. These functions give you a quick view of variable types and basic statistics, which is essential before any cleaning work. Printing the first ten rows with `head(messy_sales, 10)` provides a tangible glimpse of the raw records, helping you spot obvious formatting issues. This step shows you are following a logical data-exploration workflow.

Business Application

Although you have not yet documented the business impact of missing values or outliers, the assignment expects you to link data quality decisions to outcomes such as average sales calculations or forecasting accuracy. Adding that connection would strengthen the relevance of your work. When you eventually choose a final cleaned dataset, consider explaining how preserving sample size versus

improving data integrity affects decisions like inventory planning or marketing budget allocation. This would demonstrate practical business acumen. Future reflections should tie the technical steps (e.g., imputation method) to specific business contexts, such as how median imputation might be suitable for skewed revenue data but not for binary churn indicators.

Learning Demonstration

Your code shows you understand the mechanics of importing data and performing basic exploratory commands in R, which is a solid foundation for more advanced cleaning techniques. The placeholders for missing-value treatment and outlier handling indicate that you have identified the next learning steps: applying imputation functions (e.g., `mutate` with `ifelse(is.na(...))`) and using IQR-based filtering or winsorization. By completing the sections on imputation strategy comparison and trade-off analysis, you will demonstrate mastery of both statistical reasoning and business decision-making.

Areas for Development

Fill in all reflection question responses with detailed, example-driven explanations; this will improve both your reflection_quality score and your ability to communicate analytical reasoning. Implement at least two missing-value strategies (removal and imputation) and compare their effects on key summary statistics; documenting the before-and-after results will enhance your data_interpretation and methodology_appropriateness. Add visualizations for outlier detection (boxplots, histograms) and describe what the plots reveal about the Sales_Amount distribution; this will make your analytical narrative more compelling and business-focused.

Recommendations for Future Work

Continue practicing data cleaning pipelines in R using the `tidyr` and `dplyr` verbs; try applying them to a different public dataset (e.g., the Titanic or Ames housing data) to build fluency. Explore the `mice` package for multiple imputation and the `DescTools::Winsorize` function for capping; these tools will give you more sophisticated options when handling missing values and outliers. Develop a habit of writing a brief narrative after each code block that explains why you chose a particular method and what the results mean for the business problem at hand. This will improve both your communication clarity and your ability to justify analytical decisions.

Technical Analysis

Code Strengths:

- Your implementation of the data import and initial inspection successfully loads the messy_sales dataset and displays key information about its structure. You correctly use read_csv() from the tidyverse package and properly display row and column counts, which shows good understanding of basic data exploration techniques. The use of print() statements to show dataset overview is appropriate for this introductory assignment.
- Your approach to calculating missing values using colSums(is.na()) is correct and produces accurate results. You properly identify total missing values and missing values per column, which demonstrates solid understanding of data quality assessment. The way you display incomplete rows using the subset approach shows good logical thinking about identifying problematic data entries.
- Your implementation of the mode function using unique(), tabulate(), match(), and which.max() is well-structured and correctly identifies the most frequent value in categorical data. You

appropriately apply this function to impute missing Customer_Name values, which shows understanding of how to handle missing categorical data through mode imputation.

Code Improvement Suggestions:

• While your use of base R for missing value calculations works correctly, you could also consider using dplyr's summarise() and across() functions to make the missing value analysis more concise: colSums(is.na(messy_sales)) and sum(is.na(messy_sales)) could be simplified with summarise(across(everything(), ~sum(is.na(.x)))). This alternative approach offers cleaner syntax and is commonly used in professional analytics work.

• Your outlier detection using IQR method is well-executed, but you could enhance the boxplot creation by adding more descriptive labels: ggplot(sales_imputed, aes(y = Sales_Amount)) + geom_boxplot() + labs(title = 'Sales Amount Distribution', y = 'Sales Amount') + theme_minimal(). This would make your visualization more professional and informative for business stakeholders.

• An alternative approach for imputing numeric data like Quantity would be to use the median() function directly with na_if() to handle missing values: sales_imputed$Quantity <- ifelse(is.na(sales_imputed$Quantity), median(sales_imputed$Quantity, na.rm = TRUE), sales_imputed$Quantity). This technique offers cleaner syntax and is commonly used in data cleaning workflows.

Technical Observations:

• You demonstrate solid understanding of data quality assessment through your comprehensive approach to identifying missing values, outliers, and data inconsistencies. Your ability to systematically analyze data structure and identify potential issues shows good analytical thinking that's essential for business analytics work. This foundation will serve you well as you tackle more complex data cleaning challenges.

• Your appropriate use of base R functions for data manipulation and analysis shows good technical competency in fundamental programming concepts. You correctly apply functions like colSums(), is.na(), and logical subsetting to achieve your data cleaning objectives. This demonstrates developing expertise in reproducible data analysis workflows that are crucial for business applications.

• Your code organization supports reproducible analysis through clear sectioning and logical flow from data import to final dataset selection. You structured the missing value treatment section in a way that makes the logic clear and the results verifiable. This attention to code quality and organization is important for professional analytics work and shows good programming practices.

Additional Code Enhancement Examples:

Data Exploration Enhancement:

```
# More comprehensive data inspection
```

```
glimpse(sales_df) # dplyr alternative to str()
```

```
skimr::skim(sales_df) # Detailed summary statistics
```

```
DataExplorer::plot_missing(sales_df) # Visualize missing data
```

Data Visualization:

```
# Basic plots for data exploration
```

```
ggplot(sales_df, aes(x = amount)) + geom_histogram()
```

```
ggplot(sales_df, aes(x = category, y = amount)) + geom_boxplot()
```

Data Cleaning:

```
# Handle missing values
```

```
sales_df <- sales_df %>%
```

```
filter(!is.na(amount)) %>%
```

```
mutate(amount = ifelse(amount < 0, 0, amount))
```

Performance by Category