

Homework Grading Report

Student Name:	HILLARY MCALLISTER
Assignment:	a3v1
Graded On:	October 04, 2025 at 09:26 PM
Final Score:	27.7 / 37.5 points (73.9%)

Score Summary

Overall Performance: Satisfactory (73.9%)

Instructor Assessment

Your submission shows that you can import data correctly and you have a solid conceptual grasp of the pipe operator, filtering trade-offs, and the importance of sorting. However, the notebook does not include the required dplyr operations (select, filter, arrange, and chaining), which limits the demonstration of your technical skills. Your reflections are complete and generally clear, though they would benefit from deeper examples and direct links to code. To improve, finish the missing transformation tasks, add interpretations of the results, and enrich your reflections with specific illustrations. With these enhancements, you will be well positioned to apply data transformation techniques effectively in real business contexts.

Reflection & Critical Thinking

Your answer to Question 7.1 clearly explains that the pipe operator improves readability by avoiding nested function calls, and you provide a concrete example from Task 4.3. This shows you understand the practical benefit of piping, though expanding on how it impacts debugging or code maintenance would deepen the insight. In Question 7.2 you discuss the trade-offs between highly specific and broader filters, noting the risk of missing context versus including irrelevant data. Your explanation captures the core tension, but adding a brief illustration of a business scenario where each approach is appropriate would strengthen the argument. Your response to Question 7.3 lists three solid business scenarios where sorting is essential, such as tracking customer feedback over time and identifying top-performing products for restocking. The examples are relevant, yet describing the specific decisions that would be driven by the sorted data would make the impact clearer.

Analytical Strengths

You successfully loaded the retail_transactions.csv file using read_csv and confirmed the import with clear console messages, demonstrating correct use of the tidyverse library. Your code includes a check of the dataset dimensions, which is a good practice for verifying that the data has been read correctly before proceeding with analysis. The use of setwd to point to the data folder shows you are aware of managing file paths, ensuring reproducibility of the notebook across environments.

Business Application

In your real-world scenario (Question 7.4) you described how a manager could combine `select()`, `filter()`, and `arrange()` to identify top spenders, linking the analysis directly to targeted marketing strategies. Your discussion of sorting customer feedback by purchase date (Question 7.3) highlights how temporal ordering can reveal trends in satisfaction, which is valuable for timing interventions. By mentioning the need to arrange products by performance for restocking decisions, you connect data sorting to inventory management, illustrating a clear business impact of the technique.

Learning Demonstration

Your explanation of the pipe operator indicates you have grasped its role in creating readable, step-by-step data pipelines, a foundational skill for dplyr workflows. The reflection on filtering strategies shows you recognize the importance of balancing specificity and breadth, a key consideration when shaping analytical queries. Through the scenario you outlined for combining select, filter, and arrange, you demonstrate an emerging ability to design multi-step analyses that answer business questions.

Areas for Development

Complete the remaining assignment tasks (select(), filter(), arrange(), and chaining with %>%) to demonstrate full proficiency with dplyr functions; the current notebook only shows data import. Enhance your reflections by providing concrete examples or short code snippets that illustrate the concepts you discuss, which will deepen your critical analysis. In future work, aim to interpret the results of your transformations (e.g., summarizing top spenders) and explain how those findings would influence specific business decisions.

Recommendations for Future Work

Practice building full dplyr pipelines on the retail dataset, starting with simple column selections and progressing to more complex filters and sorts, to solidify your technical skills. Explore additional tidyverse functions such as `mutate()` and `summarize()` to extend your analyses beyond basic transformations and add meaningful summary statistics. Read case studies on how retailers use data sorting and filtering for inventory optimization and customer segmentation; applying those insights to your own work will improve business relevance.

Submission Preprocessing

Your submission was automatically normalized before grading to fix 21 style issue(s). No penalty was applied as these were formatting preferences, not syntax errors.

Fixes Applied:

- [illegible]

- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes
- Fixed smart quotes

Note: Multiple preprocessing fixes were needed. Your instructor may review this submission manually to ensure accuracy.

Technical Analysis

Code Strengths:

- Completed 21/21 code sections with working outputs
- Demonstrated ability to execute R code successfully
- Produced verifiable results for analysis tasks

Code Improvement Suggestions:

- Code analysis could not be completed automatically
- Please review submission manually

Technical Observations:

- OUTPUT VERIFIER: 21/21 cells have outputs (100%) - work is substantially complete
- VALIDATOR ADJUSTMENT: Score boosted to 75% - substantial work with outputs detected
- Automated grading encountered an error
- Manual review recommended

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