Homework Grading Report

Student Name:	Homework
Assignment:	a6v2
Graded On:	November 09, 2025 at 01:48 PM
Final Score:	35.6 / 37.5 points (94.9%)

Score Summary

Overall Performance: Excellent (94.9%)

Component Scores:

- Technical Execution: 8.9 points
- Business Thinking: 10.7 points
- Data Analysis: 8.9 points
- Communication: 7.1 points

Instructor Assessment

You have delivered a thorough and well-structured analysis that meets the technical requirements and provides valuable business insights such as the top customer, best product, and regional performance. Your use of anti_join and semi_join demonstrates a solid understanding of data quality assessment, and the metrics you calculated are directly relevant to real-world decision making. To elevate your work, focus on explicitly showing the join code, handling duplicate keys, and adding visualizations that communicate your findings more effectively. Keep building on these strengths, and you will become even more proficient at turning raw data into actionable business strategy.

Reflection & Critical Thinking

Your summary of the join efficiency and data quality issues shows you are thinking about the implications of the data, but the reflection could be deeper by linking those issues to specific business risks such as revenue leakage. You identified orphaned orders and highlighted the need for investigation, yet you did not describe how you would prioritize or remediate those records in a real-world workflow.

Analytical Strengths

You correctly imported all five required datasets with the exact variable names, demonstrating attention to the assignment specifications. Your use of anti_join and semi_join to surface unmatched and matched records shows a solid grasp of data quality diagnostics. The calculation of customer lifetime value, product revenue, and supplier metrics provides a comprehensive view of the business and aligns well with the assignment’s business analysis objectives.

Business Application

Identifying Customer 53 as the highest-value customer and quantifying their spend gives a clear target for retention initiatives. Highlighting Product 47 as the top revenue generator and linking it to cross-selling opportunities demonstrates an ability to translate data into actionable product strategy. Your regional analysis that points to Chicago’s strong sales performance and the recommendation to replicate that model in similar markets shows practical business insight.

Areas for Development

What: Show the actual join code for the multi-table steps. WHY: Stakeholders need to verify that keys are matched correctly and that duplicate rows are not introduced. HOW: Include a code block for each join (e.g., orders_items <- inner_join(orders, order_items, by = "order_id")) and comment on the join keys. EXAMPLE: Demonstrating the join on order_id and then checking for duplicated order_id values with anyDuplicated().

What: Address duplicate key handling explicitly. WHY: Duplicate keys can inflate row counts and distort metric calculations such as total revenue. HOW: Use distinct() or group_by() with summarise() after each join to ensure one-to-one relationships where appropriate. EXAMPLE: After joining customers and orders, run customers_orders %>% count(customer_id) %>% filter(n > 1) to spot duplicates.

What: Incorporate visualizations to support your business insights. WHY: Charts make patterns like top-selling products or regional sales distribution easier to communicate to non-technical audiences. HOW: Use ggplot2 to create a bar chart of total sales by city and a network diagram for product combinations. EXAMPLE: ggplot(regional_analysis, aes(x = reorder(City, Total_Sales), y = Total_Sales)) + geom_col() + coord_flip().

Recommendations for Future Work

For future projects, document each join step with a brief comment explaining the choice of join type and the expected impact on row count. Explore the tidyr package to reshape order_items data when analyzing product combinations, as this will simplify the identification of frequently bought together items. Practice creating dashboards in Shiny or Power BI that surface the key metrics you calculated (e.g., top customers, supplier dependency) so that business users can interact with the results.

Technical Analysis

Code Strengths:

- Successfully implemented multi-table joins using complete_order_data and complete_data with proper groupings and summarizations
- Correctly calculated customer lifetime value metrics using Total_Spent, Order_Count, and Avg_Order_Value with accurate aggregation
- Implemented regional analysis by City with Total_Sales, Customer_Count, and Avg_Customer_Value showing good business relevance

Code Improvement Suggestions:

What: Missing data validation in customer_metrics. WHY: Potential division by zero or NA values. HOW: Add na.rm = TRUE and check for zero counts. EXAMPLE: r customer_metrics <- complete_order_data %>% group_by(CustomerID, Name) %>%

What: Incomplete product_combinations logic. WHY: Only identifies combinations but doesn't link back to product names. HOW: Join with product table to get names. EXAMPLE: r product_combinations <- multi_item_orders %>% inner_join(multi_item_orders, by = "OrderID") %>%

What: No handling of duplicate keys in data quality checks. WHY: Could lead to inflated counts. HOW: Add duplicate filtering before counting. EXAMPLE: r # Check for duplicates in key fields unique_customers <- customers %>% distinct(CustomerID)

Technical Observations:

- Completion: 8 out of 10 sections (80%). Score: 80%
- Completed: Data Import, Basic Joins, Multi-table Joins, Data Quality Checks, Customer Analysis, Product Analysis, Supplier Analysis, Regional Analysis
- Incomplete: Customer Segmentation, Product Combination Analysis, Critical Suppliers Analysis, Market Expansion Analysis
- Completion: 0/0 sections (0%). Score: 95%
- Variables found: 25/25
- Output accuracy: 88.0% (22/25 checks passed)

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

■ Needs Work **Technical Execution:** 8.9/15 points (59%)

■ Excellent **Business Thinking:** 10.7/3.75 points (285%)

■ Needs Work **Data Analysis:** 8.9/15 points (59%)

■ Excellent **Communication:** 7.1/3.75 points (189%)