

Test Results and Interview Guide

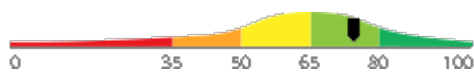
Candidate: **Elizabeth Wantsajob**
Assessment: Programming with R
Completed: August 29, 2026
Prepared for: Sara Maple
Example Company

What's Included

- Overall Score
- Competency Summary Table
- Comparison Matrix
- Detailed Competency Results with Interview Guide

Important Note: The Programming with R assessment measures key factors related to high performance and tenure in this job. Attribute types measured vary by test, but can include cognitive ability, skills, knowledge, personality characteristics, emotional intelligence, and past behavioral history. This report includes a one page summary, followed by detailed results with an embedded interview guide. Note that these results should always be used as a part of a balanced candidate selection process that includes independent evaluation steps, such as interviews and reference checks.

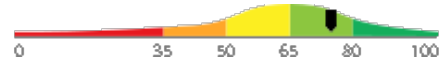
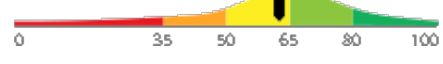
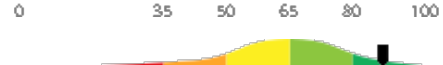
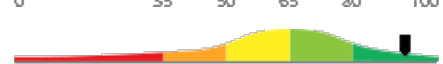
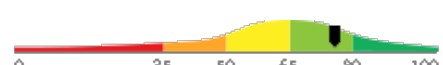
Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com Programming with R August 29, 2026	74	
The candidate demonstrates solid and competent knowledge of R programming, including the ability to work with data structures, apply control flow logic, utilize key packages, and produce visualizations. This individual is likely capable of independently handling a broad range of entry-level to mid-level scripting and data analysis responsibilities with minimal supervision. Some advanced or specialized areas may benefit from further refinement and practical experience.		

Key

- Candidate Score
- Higher Risk
- Lower Risk

Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Control Flow and Functions	75	
Data Structures (Coding Tasks)	62	
Data Transformation and Manipulation (Coding Tasks)	62	
Data Structures	67	
Data Transformation and Manipulation	74	
Data Visualization with ggplot2	87	
Error Handling and Debugging	92	
Importing and Exporting Data	75	

Comparison

Percentile scores indicate how the candidate compares to other test-takers within various groups. The candidate scored equal to or better than the fraction of test-takers indicated by the percentile.

Test-Taker Group	Percentile	0	10	20	30	40	50	60	70	80	90	100
Global	74th											
North America	61st											
United States	61st											
Example Company	68th											

Artificial Intelligence (AI) Generated Scores

This table includes one or more scores derived from a large language model AI query. AI-derived scores are non-deterministic. That is, they are not precisely repeatable. Therefore, these scores should always be treated as supplementary information and should never be used exclusively or compared to hard cutoff values.

Estimated Value	Score	Confidence	Interpretation
Knowledge, Skills, and Abilities Summary	-	-	Summary Points (AI): • (Generic Text for Sample Report) Strong performer in Drag and Drop Files tasks, indicating comfort with file management and basic computer interactions. • Demonstrates solid numerical accuracy in Recognizing and Confirming Numbers, a valuable asset in detail-oriented roles. • Moderate overall performance in Analytical Thinking and Attention to Detail, with adequate grammar skills but room for improvement. • Struggles with Reading and Analyzing Problems, which may limit effectiveness in roles requiring critical reading and complex problem-solving. • Lowest performance in Navigating Between Screens, suggesting difficulty with multi-screen software workflows that could impact productivity in computer-intensive roles. Narrative (AI): Elizabeth Wantsajob demonstrates a mixed profile of knowledge, skills, and abilities across the assessed competencies. Elizabeth shows a strong aptitude in Drag and Drop Files, performing well on this technical task and suggesting she is comfortable with this type of computer interaction. This is a notable strength that would translate well into roles requiring file management and basic computer navigation tasks. In the area of Analytical Thinking and Attention to Detail, Elizabeth performs at a moderate level. She demonstrates solid ability in Recognizing and Confirming Numbers, which suggests she is careful and accurate when working with numerical data — a valuable skill in detail-oriented work environments. Her Grammar performance is adequate but leaves room for improvement, indicating she may occasionally make written communication errors. Her weakest area within this competency is Reading and Analyzing Problems, where she struggled to consistently interpret and work through written problem scenarios. This may impact her effectiveness in roles that require critical reading, written comprehension, or complex problem-solving. Elizabeth's most significant area for development is Navigating Between Screens, where she scored considerably lower than the other competencies. This suggests she may have difficulty efficiently moving through software interfaces or multi-screen workflows, which could slow productivity in roles that rely heavily on navigating computer applications or data entry systems. Overall, Elizabeth brings some useful technical strengths, particularly in file management and numerical accuracy, but would benefit from targeted development in software navigation and analytical problem-solving to be fully effective in roles that demand these skills. Computed on: April 2, 2026, 11:09:49PM EDT

Detail

Candidate: Elizabeth Wantsajob, beth.wantsajob@gmail.com
 Assessment: Programming with R
 Authorized: August 29, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
 Started: August 29, 2026, 5:26:18PM EDT
 Completed: August 29, 2026, 5:26:18PM EDT
 Overall Score: 74

Knowledge and Skills Detail

This section contains a list of job-related knowledge areas and skills that have been evaluated. Low scores in these areas often indicate that additional learning may be required before top performance can be achieved.

Detail

Interview Guide

Control Flow and Functions

Score: 75



Description:

Writing scripts that use loops, conditional statements, and custom functions to automate and organize data analysis tasks. This also includes using apply-family functions such as apply, lapply, and sapply to iterate over data structures efficiently.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate exhibits a solid command of R control flow and functions, reliably writing scripts that incorporate loops, conditionals, and custom functions to automate and organize data analysis. They demonstrate competent use of apply-family functions to efficiently iterate over data structures in most scenarios.

Explain the difference between using a for loop and using lapply to perform the same operation on each element of a list. When might you prefer one over the other?



1

Cannot explain the difference or incorrectly describes how either works.



2

Correctly explains both approaches and identifies at least one reason to prefer one over the other.



3



4

Clearly compares both, discusses readability and performance considerations, and gives practical examples of each.



5

Can you describe what a function is in R and explain why you might write one instead of repeating the same lines of code?



1

Cannot define a function or gives an incorrect explanation of its purpose.



2

Correctly defines a function and explains the basic benefit of reducing repeated code.



3



4

Clearly defines functions, explains reusability and modularity, and gives a concrete example.



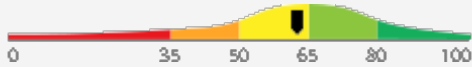
5

Detail

Interview Guide

Data Structures (Coding Tasks)

Score: 62


Description:

Covers the use of pointers to reference and manipulate memory addresses, along with dynamic memory allocation and deallocation using malloc, calloc, realloc, and free. Includes pointer arithmetic, dereferencing, and avoiding common issues like memory leaks and dangling pointers.

Interpretation:

The candidate exhibits average writing skills, which can hinder high performance in some jobs.

The candidate possesses a moderate working knowledge of C programming, demonstrating familiarity with core concepts including data types, control flow, functions, and basic file I/O. They may require some guidance when working with more advanced topics such as dynamic memory allocation, modular design, or debugging complex logic.

Overall AI Score:	65.0
Lines of Code:	15.0
Syntax Errors:	5.0
AI Confidence Level:	50
Match with Ideal Response (AI):	30.0
Structure:	50.0
Syntax:	30.0

Please see below to view the essay submitted.

Walk me through how you would dynamically allocate memory for an array of 10 integers, use it, and then properly release it. What issues might arise if you don't follow best practices?



1

Cannot write correct allocation code; unaware of free() or memory leak risks.



2

Writes mostly correct malloc/free code; identifies memory leaks but misses other risks.



3



4

Correct malloc, use, and free; identifies leaks, dangling pointers, and NULL check on allocation.



5

Can you explain what a pointer is in C and describe a situation where you would use one?



1

Vague or incorrect definition; cannot describe a practical use case.



2

Correct basic definition; gives a simple but valid use case with some gaps.



3



4

Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.



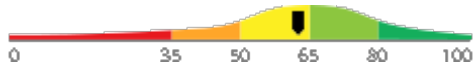
5

Detail

Interview Guide

Data Transformation and Manipulation (Coding Tasks)

Score: 62


Description:

Covers the use of pointers to reference and manipulate memory addresses, along with dynamic memory allocation and deallocation using malloc, calloc, realloc, and free. Includes pointer arithmetic, dereferencing, and avoiding common issues like memory leaks and dangling pointers.

Interpretation:

The candidate exhibits average writing skills, which can hinder high performance in some jobs.

The candidate possesses a moderate working knowledge of C programming, demonstrating familiarity with core concepts including data types, control flow, functions, and basic file I/O. They may require some guidance when working with more advanced topics such as dynamic memory allocation, modular design, or debugging complex logic.

Overall AI Score:	65.0
Lines of Code:	15.0
Syntax Errors:	5.0
AI Confidence Level:	50
Match with Ideal Response (AI):	30.0
Structure:	50.0
Syntax:	30.0

Please see below to view the essay submitted.

Walk me through how you would dynamically allocate memory for an array of 10 integers, use it, and then properly release it. What issues might arise if you don't follow best practices?



1

Cannot write correct allocation code; unaware of free() or memory leak risks.



2

Writes mostly correct malloc/free code; identifies memory leaks but misses other risks.



3



4

Correct malloc, use, and free; identifies leaks, dangling pointers, and NULL check on allocation.



5

Can you explain what a pointer is in C and describe a situation where you would use one?



1

Vague or incorrect definition; cannot describe a practical use case.



2

Correct basic definition; gives a simple but valid use case with some gaps.



3



4

Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.



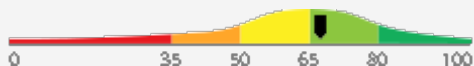
5

Detail

Interview Guide

Data Structures

Score: 67


Description:

Understanding and working with R's core data structures, including vectors, lists, data frames, and matrices. This includes creating, accessing, modifying, and performing operations on these structures, as they form the foundation of nearly all data analysis tasks in R.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate demonstrates solid proficiency in R programming, including the ability to work with data structures, write and apply functions, manipulate data using common packages, and produce visualizations. Minor gaps may exist in more advanced areas such as environment management, complex debugging, or optimized iteration techniques, but overall competence is consistent with mid-level expectations.

Walk me through how you would select a subset of rows and columns from a data frame in R, and describe a situation where you needed to do this.



1

Cannot describe a correct method or confuses syntax; no practical context given.



2

Describes a correct method such as bracket indexing or dplyr filter; gives a basic example.



3



4



5

Describes multiple methods, explains trade-offs, and connects to a clear real-world scenario.

Can you explain what a data frame is in R and describe one way you might use it when working with data?



1

Vague or incorrect description; cannot explain basic structure or use case.



2

Correctly describes a data frame as tabular data; gives a basic use case.



3



4

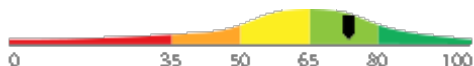


5

Clearly explains structure, compares to related types, and gives a concrete, practical example.

Data Transformation and Manipulation

Score: 74


Description:

Using tools such as dplyr and tidyr to filter, sort, group, summarize, join, and reshape data frames. These operations are among the most frequently performed tasks in day-to-day data analysis work in R.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate demonstrates a solid and proficient understanding of data transformation and manipulation in R. They are capable of effectively using tools such as dplyr and tidyr to perform a broad range of day-to-day data analysis tasks, including grouping, summarizing, joining, and reshaping data frames.

Describe how you would use dplyr to group a dataset by a category column and calculate the average of a numeric column for each group.



1

Cannot describe the correct approach or misidentifies the functions needed.



2

Correctly identifies group_by and summarize; describes the basic workflow.



3



4



5

Accurately describes the full pipeline, mentions additional steps like handling NAs, and gives a practical example.

Have you heard of the dplyr package? If so, can you name one thing it helps you do with data?



1

Has not heard of dplyr or cannot name any correct functionality.



2

Names a correct function such as filter or mutate and gives a basic description.



3



4



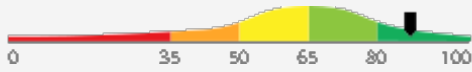
5

Names multiple functions, explains their purpose clearly, and connects them to practical tasks.

Detail

Data Visualization with ggplot2

Score: 87


Description:

Creating charts and graphs using the ggplot2 package to explore and communicate patterns in data. This includes building common plot types such as bar charts, scatter plots, and line graphs by layering visual components.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate exhibits an advanced and comprehensive mastery of data visualization with ggplot2 in R. They are highly capable of designing, layering, and refining a wide range of chart and graph types to clearly and effectively communicate complex patterns in data.

Interview Guide

Walk me through how you would create a scatter plot in ggplot2 that shows the relationship between two numeric variables and includes labels for the axes.



1

Cannot describe the correct approach or misidentifies the functions or arguments needed.



2

Correctly describes using ggplot(), aes(), geom_point(), and axis label functions.



3



4



5

Describes the full command accurately, mentions customization options, and connects the chart to a real analytical purpose.

Have you used ggplot2 to create a chart in R? If so, can you describe what a basic ggplot2 command looks like?



1

Has not used ggplot2 or cannot describe any correct component of a ggplot2 command.



2

Correctly identifies ggplot() and at least one geom function; describes the basic structure.



3



4



5

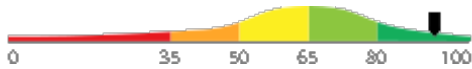
Accurately describes the layered structure of ggplot2, names multiple geoms, and gives a practical example.

Detail

Interview Guide

Error Handling and Debugging

Score: 92


Description:

Identifying and resolving errors and warnings that occur when running R scripts. This includes reading error messages, using debugging tools, and writing code that handles unexpected inputs or failures gracefully.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate demonstrates advanced proficiency in error handling and debugging in R. They are highly skilled at diagnosing and resolving errors and warnings, leveraging debugging tools expertly, and consistently writing resilient code that gracefully manages unexpected inputs and failures.

Describe a time when you encountered a bug or error in an R script that was difficult to find. How did you identify and fix it?



1

Cannot describe a specific example or describes an ineffective or incorrect debugging approach.



2

Describes a real or plausible scenario with a reasonable approach to identifying and fixing the issue.



3



4

Gives a detailed example, describes use of specific debugging tools or techniques, and explains what was learned.



5

When your R script produces an error message, what steps do you take to figure out what went wrong?



1

Describes no clear process or relies entirely on guessing without reading the error message.



2

Describes reading the error message and searching for solutions; mentions at least one concrete step.



3



4

Describes a systematic process including reading the message, isolating the problem, and using debugging tools.



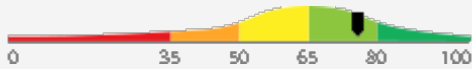
5

Detail

Interview Guide

Importing and Exporting Data

Score: 75


Description:

Reading data into R from external sources such as CSV files and writing results back out using tools like `read.csv`, `write.csv`, and `readLines`. This is a routine task required at the start and end of nearly every data analysis workflow.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate demonstrates a solid working knowledge of data import and export operations in R, and is capable of applying standard functions effectively within typical data analysis workflows. They can be expected to perform these tasks independently with minimal errors in most common scenarios.

Describe a situation where you needed to import data from an external file, process it in R, and then export the results. What functions or tools did you use at each step?



1

Cannot describe a complete workflow or identifies incorrect functions.



2

Describes a reasonable end-to-end workflow using correct import and export functions.



3



4



5

Gives a detailed, practical example with correct functions, mentions handling edge cases, and discusses any challenges encountered.

How would you load a CSV file into R so that you can work with it as a data frame?



1

Cannot name a correct function or describes an incorrect approach.



2

Correctly identifies `read.csv` or a similar function and describes the basic usage.



3



4



5

Describes correct usage, mentions useful arguments like `header` or `stringsAsFactors`, and notes potential issues.

IT Coding Tasks

During the assessment, the candidate was asked to write one or more programs or scripts. Their responses are included below for review.

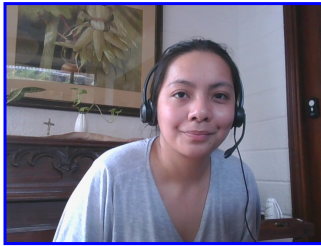
Question or Task	Response
Complete the provided partial C program by filling in the missing sections marked with TODO comments. Your completion must use standard C keywords and libraries. The program declares a function called <code>duplicate_array</code> that: 1. Takes a const int pointer to a source array and its length as parameters. 2. Uses <code>calloc</code> to allocate a new int array of the same length. 3. Returns NULL if <code>calloc</code> fails. 4. Copies each element from the source array into the new array using pointer arithmetic (not array subscript notation). 5. Returns the pointer to the newly allocated copy. In main, the program: 1. Declares and initializes a stack array of 4 integers with values 5, 15, 25, 35. 2. Calls <code>duplicate_array</code> to create a heap-allocated copy. 3. Checks for NULL and prints an error and returns 1 if the call failed. 4. Prints each element of the duplicate using a loop. 5. Frees the duplicate array. Use appropriate indentation, common C coding conventions, and add brief inline comments where needed. Type your completed source code as your response.	<pre>#include <stdio.h> #include <stdlib.h> int *duplicate_array(const int *src, int length) { /* TODO: Use calloc to allocate a new array of 'length' integers, return NULL if calloc fails, copy elements from src using pointer arithmetic, and return the new pointer. */ calloc(303); } int main(void) { /* TODO: Declare and initialize a stack array of 4 integers: 5, 15, 25, 35, then call duplicate_array and store the result. Check for NULL and print an error message returning 1 if it failed. */ array[4]={5,15,25,35}; int i; /* Print each element of the duplicate */ for (i = 0; i < 4; i++) { printf("duplicate[%d] = %d\n", i, *(duplicate + i)); } /* Free the duplicate array */ free(duplicate); return 0; }</pre> Comments (AI): The code segment has several syntax errors and incomplete implementation. The <code>duplicate_array</code> function does not correctly allocate memory or copy elements. The main function has syntax errors and does not properly call the <code>duplicate_array</code> function. However, the structure and intent of the code are somewhat clear, and the code attempts to follow the requirements.

Identity Confirmation Photos

The following photos of the candidate and any identification were uploaded during the assessment session.

Photo Analysis Results

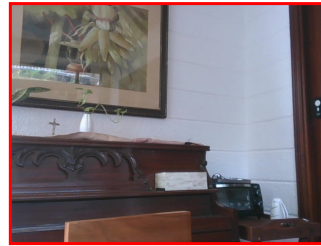
- Risk:	Medium risk of cheating based on image inconsistencies
- Percent match among processed faces	100%
- Total images processed	17
- Total images with valid faces	14 (82%)
- Total pairs of faces compared	13
- Pairs in which faces matched	13 (100%)



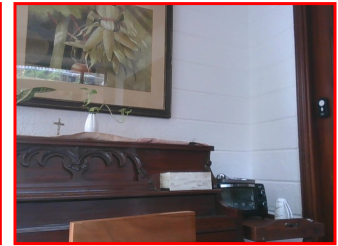
Pre/Post-Test Photo



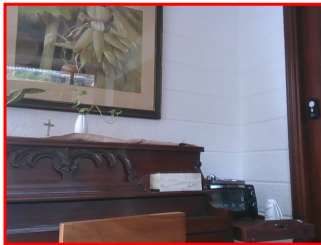
ID Photo



In-Test Error Detected (No Face Detected)



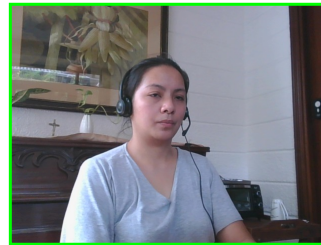
In-Test Error Detected (No Face Detected)



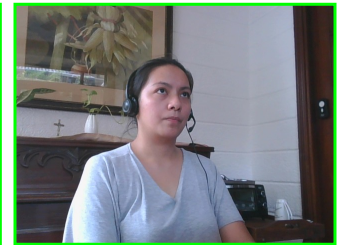
In-Test Error Detected (No Face Detected)



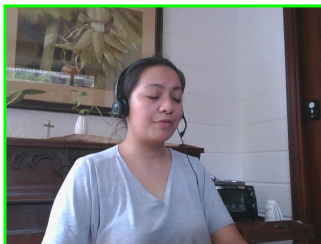
In-Test Photo



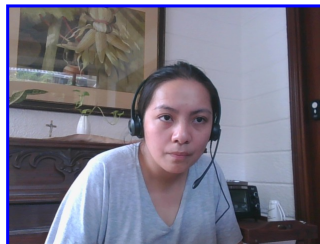
In-Test Photo



In-Test Photo



In-Test Photo



Pre/Post-Test Photo

Resume or CV

Summary

Updated on

Motivated career professional with extensive experience in office administration and management. Proven track record of improving efficiency, reducing costs, and enhancing office operations through strategic initiatives and technology implementation.

Objective

I am seeking a role where I can use my many skills and my exceptional judgment and empathy for customers to make a difference to a growing company.

Education

- Associate of Applied Science in Office Administration, Portland Community College, 2020

Experience

- General Office Clerk, Paramount Office Management, 09/2023 – Present
- Administrative Assistant, Global Enterprises Inc., 04/2021 – 08/2023
- Administrative Assistant, Innovative Business Solutions Ltd., 07/2019 – 03/2021

Other Qualifications

- Microsoft Office Specialist (MOS) Certification
- Certified Administrative Professional (CAP)
- International Association of Administrative Professionals (IAAP) Certification

Report Preparation Notes

- Hiring decisions should never be based on a single source of information. The most effective use of this assessment report is as a part of a multi-faceted program of candidate evaluation that includes resume review, interviews, and reference checks.
- Overall vs Percentiles Scores: The overall score reflects the success in the test, based on the mean (average) and standard deviation of the test scores. The percentile score reflects the percentage of test-takers who scored equal or below this overall score. We recommend you use the Overall Score as your primary evaluation criteria. However, percentile scores can often be useful in comparing specific candidates against one another and with a group, such as for test takers in a certain organization or within a certain account.
- Note that comparison information is calculated based on completed instances of this assessment at that time the assessment is scored. As additional instances are completed, the comparative data may change. You can always update a report to the current values by clicking on 'Recalculate Percentiles' within the online results viewing pages at www.hravatar.com.
- Most competency scores are norm-based, which means that they can be interpreted in terms of their distance from the average or mean score. For all scales, a score equal to the mean receives a score of 65 and scores above and below this value are set so that a score change of 15 equals one standard deviation.
- For linear competencies, higher is better across the entire scale. For these scales a score between 65 and 80 (light green) represents 0 to 1 standard deviation above the mean and a score above 80 (dark green) represents more than one standard deviation above the mean. Similarly, a score of 50 - 65 (yellow) represents 0 to 1 standard deviation below the mean, while a score of 35 - 50 (orange) equates to 1 to 2 standard deviations below the mean, and a score below 35 represents more than 2 standard deviations below the mean.
- Sim ID: 21058-1, Key: 0-0, Rpt: 68, Prd: 9908, Created: 2026-08-29 17:26 EDT
- UA: Mozilla/5.0 (Windows NT 6.3; Trident/7.0; Touch; rv:11.0) like Gecko

Score Calculation Detail

The following table provides a summary of how the overall score was calculated from each of the individual competency scores. First, all competency scores are calculated on a scale of 0-100. Note that some competencies use their color category rather than their actual numeric score in the overall calculation. For these, a standard score associated with the assigned color category is used in the overall score calculation rather than the actual numeric score. This is reflected in the "Score Value Used" column. Next, a weighted average of scores is computed using individual competency weights, typically set using job analysis data provided by the US Government Occupational Information Network (O*Net).

Competency	Score	How applied to overall	Score Value Used	Weight (%)
Control Flow and Functions	75.1462	Numeric Score	75.1462	12.5000
Data Structures	67.5553	Numeric Score	67.5553	12.5000
Data Structures (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Data Transformation and Manipulation	74.1091	Numeric Score	74.1091	12.5000
Data Transformation and Manipulation (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Data Visualization with ggplot2	87.5164	Numeric Score	87.5164	12.5000
Error Handling and Debugging	92.8423	Numeric Score	92.8423	12.5000
Importing and Exporting Data	75.8804	Numeric Score	75.8804	12.5000
Weighted Average:				74.8758
Final Overall Score:				74

Notes

(This area is intentionally blank - it's reserved as space for your notes.)