MAJOR COURSE OUTPUT #2: DATA ANALYSIS PIPELINE FOR FLOOD CONTROL PROJECTS

INTRODUCTION

This major course output presents the development of a Data Analysis Pipeline that demonstrates fundamental concepts in programming paradigms and data processing. The application is designed to ingest a real-world CSV dataset on DPWH flood control projects, perform preprocessing, and generate three tabular reports to facilitate analysis of infrastructure trends, financial efficiencies, and performance metrics. **A key component of this project is the use of libraries or packages to process the data.**

FUNCTIONAL SPECIFICATIONS

MANAGING DATA INGESTION

REQ #	DETAILS
REQ-0001	Provision to read the CSV file dpwh_flood_control_projects.csv containing 9,800+ rows of flood mitigation projects.
REQ-0002	Provision to perform basic validation: Log total row count and detect/parse errors (e.g., invalid dates or missing values).
REQ-0003	Provision to filter projects from 2021-2023 (exclude 2024 entries for analysis stability).
REQ-0004	Provision to compute derived fields: • CostSavings = ApprovedBudgetForContract - ContractCost; • CompletionDelayDays = days between StartDate and ActualCompletionDate (positive if delayed).
REQ-0005	Provision to clean data uniformly: • Convert financial fields to floats (PHP); • parse dates or use date data types when possible;

	• impute or filter incomplete rows (e.g., null lat/long via provincial averages).

MANAGING REPORT GENERATION

REQ #	DETAILS
REQ-0006	Provision to generate Report 1: Regional Flood Mitigation Efficiency Summary. This table will have the following columns: • aggregate total `ApprovedBudgetForContract`, • median `CostSavings`, • average `CompletionDelayDays`, and • percentage of projects with delays >30 days by `Region` and `MainIsland`. Include "Efficiency Score", which is computed as: (median savings / average delay) * 100, normalized to 0-100. Output as sorted CSV (descending by EfficiencyScore).
REQ-0007	Provision to generate Report 2: Top Contractors Performance Ranking. Rank top 15 `Contractor`s by total `ContractCost` (descending, filter >=5 projects), with columns for the following: • number of projects, • average `CompletionDelayDays`, • total `CostSavings`, • "Reliability Index", which is computed as (1 - (avg delay / 90)) * (total savings / total cost) * 100 (capped at 100). Flag <50 as "High Risk". Output as sorted CSV.
REQ-0008	Provision to generate Report 3: Annual Project Type Cost Overrun Trends. Group by `FundingYear` and `TypeOfWork`, computing the following: • total projects • average `CostSavings` (negative if overrun) • overrun rate (% with negative savings) • year-over-year % change in average savings (2021 baseline).

	Output as sorted CSV (ascending by year, descending by AvgSavings).
REQ-0009	Provision to produce a `summary.json` aggregating key stats across reports (e.g., total number of projects, total number of contractors, total provinces with projects, global average delay, total savings).

TECHNICAL SPECIFICATION

REQ #	DETAILS
REQ-0010	Application should be developed / built on the following programming languages: ● R ● JavaScript ● Kotlin ● Rust
REQ-0011	Provision for output standardization: Generate identical CSV files for each report (comma-formatted numbers, rounded to 2 decimals); one run command per language (e.g., `Rscript main.R`, `node index.js`).

SAMPLE OUTPUT

```
Select Language Implementation:
[1] Load the file
[2] Generate Reports

Enter choice: 1
Processing dataset... (9,852 rows loaded, 9,234 filtered for 2021-2023)

Select Language Implementation:
[1] Load the file
[2] Generate Reports

Enter choice: 2

Generating reports...
Outputs saved to individual files…

Report 1: Regional Flood Mitigation Efficiency Summary

Regional Flood Mitigation Efficiency Summary
(Filtered: 2021-2023 Projects)

| Region                          | MainIsland | TotalBudget   | MedianSavings | AvgDelay | HighDelayPct | EfficiencyScore |
|---------------------------------|------------|---------------|---------------|----------|--------------|-----------------|
| Cordillera Administrative Region | Luzon      | 1,234,567,890 | 1,234.56      | 25.3     | 15.20        | 48.75           |
| Region XIII                     | Mindanao   | 987,654,321   | 987.65        | 45.2     | 35.40        | 21.85           |

(Full table exported to report1_regional_summary.csv)

Report 2: Top Contractors Performance Ranking

Top Contractors Performance Ranking
(Top 15 by TotalCost, >=5 Projects)

| Rank | Contractor                          | TotalCost   | NumProjects | AvgDelay | TotalSavings | ReliabilityIndex | RiskFlag |
|------|-------------------------------------|-------------|-------------|----------|--------------|------------------|----------|
| 1    | ASC CONSTRUCTION & CONCRETE PRODUCTS | 500,000,000 | 15          | 30.5     | 2,500,000    | 75.20            | Low Risk |
| 2    | GICAR CONSTRUCTION, INC.            | 400,000,000 | 12          | 15.2     | 1,800,000    | 88.50            | Low Risk |

(Full table exported to report2_contractor_ranking.csv)

Report 3: Annual Project Type Cost Overrun Trends

Annual Project Type Cost Overrun Trends
(Grouped by FundingYear and TypeOfWork)

| FundingYear | TypeOfWork                                | TotalProjects | AvgSavings | OverrunRate | YoYChange |
|-------------|-------------------------------------------|---------------|------------|-------------|-----------|
| 2021        | Construction of Flood Mitigation Structure | 1,200         | 1,500.00   | 12.50       | 0.00      |
| 2021        | Construction of Revetment                 | 800           | -250.75    | 25.30       | 0.00      |
| 2022        | Construction of Flood Mitigation Structure | 1,100         | 1,200.00   | 18.20       | -20.00    |

(Full table exported to report3_annual_trends.csv)

Summary Stats (summary.json):
{"global_avg_delay": 45.2, "total_savings": 15000000}

Back to Report Selection (Y/N):
```

EVALUATION CRITERIA

Criteria	Description	Points	Details
Code Simplicity	Measures how straightforward and minimal the code is.	5	**5 pts:** Code is simple and efficient. **3-4 pts:** Mostly simple, with minor inefficiencies. **1-2 pts:** Code has unnecessary complexity. **0 pts:** Code is overly complex or unclear.
Performance	Evaluates how quickly the program executes, especially with large inputs.	5	**5 pts:** Excellent performance across all inputs. **3-4 pts:** Minor performance issues with large inputs. **1-2 pts:** Noticeable lags. 0 pts: Poor performance.
Code Readability	Assesses the clarity of the code, including formatting, variable naming, and use of comments.	5	**5 pts:** Clean, well-organized code. **3-4 pts:** Some minor readability issues. **1-2 pts:** Difficult to follow. **0 pts:** Unreadable code.
Correctness	Checks if the program produces the correct outputs and handles edge cases.	3	**3 pts:** Correct outputs for all cases. **2 pts:** Minor mistakes in edge cases. **1 pt:** Frequent errors. **0 pts:** Fails to provide correct output.
User Experience	Measures how intuitive and user-friendly the program is, including clear input/output and instructions.	2	**2 pts:** Smooth, intuitive experience. **1 pt:** Somewhat confusing interface. **0 pts:** Poor user experience.