Below is a proposed software specification that defines two Python API functions. One function is a general-purpose report generator for summarizing analysis findings, and the other is dedicated to outputting a numerical result (i.e. the count of failing dies). You can extend or adjust the parameters as needed.

**1. Report Analysis Findings**  
**Function Name**: report\_findings  
**ID**: 7\_9, 30\_9, 33\_8, 51\_8, 57\_10, 61\_7, 75\_8, 4\_9, 22\_8, 32\_8, 40\_8  
**Purpose**:  
Generate a detailed report summarizing the analysis findings for semiconductor chip test data analytics. This function is intended to collect and document various aspects of the analysis—such as statistical significance, visualizations, and recommendations—that support decision making based on logical/parametric test failure correlations, yield trends, bin-limit effects, and other quality metrics.

**Signature**:

def report\_findings(workflow\_id: int,

instruction\_id: int,

analysis\_summary: str,

statistical\_results: str = None,

recommendations: str = None,

visualizations: object = None,

additional\_context: dict = None) -> None:

"""

Generate a detailed report summarizing analysis findings for semiconductor chip test data analytics.

Parameters:

workflow\_id (int): The workflow identifier.

instruction\_id (int): The instruction identifier.

analysis\_summary (str): A textual summary of the analysis.

statistical\_results (str, optional): Details on statistical significance or related metrics.

recommendations (str, optional): Recommendations or next steps based on the analysis.

visualizations (object, optional): Visualization objects (charts, graphs) supporting the analysis.

additional\_context (dict, optional): A dictionary containing any extra context such as test measurements,

bin limits, comparative analysis details, etc.

Returns:

None

"""

pass # Implementation goes here

**Used For**:

* **Workflow 7, Instruction 9**: Report Findings – Summarize the analysis, explain statistical significance, and recommend possible next steps if strong correlations are found.
* **Workflow 30, Instruction 9**: Report Conclusions – Summarize results highlighting whether test measurements are consistent across wafer groups and if bin limits might be masking failures.
* **Workflow 33, Instruction 8**: Report Findings – Create a summary report detailing comparative analysis, statistical significance, and corrective actions if a test house is associated with lower yields.
* **Workflow 51, Instruction 8**: Report Findings – Document analysis including statistical results and visualizations regarding tester variation.
* **Workflow 57, Instruction 10**: Report Findings – Summarize analysis to highlight significant correlations and provide recommendations for further investigation.
* **Workflow 61, Instruction 7**: Report Findings – Summarize statistical trend analysis and visual observation regarding yield performance over recent lots.
* **Workflow 75, Instruction 8**: Report Findings – Summarize correlation results with context (significance, confidence levels) and further analysis recommendations.
* **Workflow 4, Instruction 9**: Report Findings – Summarize a detailed report with statistical evidence, visualizations, and conclusions on low-yield patterns related to specific test houses/testers.
* **Workflow 22, Instruction 8**: Report Generation – Summarize analysis, results, and recommendations in a detailed report.
* **Workflow 32, Instruction 8**: Report Findings – Document analysis to highlight key parametric tests associated with yield variation and provide further recommendations.
* **Workflow 40, Instruction 8**: Report Findings – Summarize analysis to determine if the current lot’s yield is within control limits or if it is an outlier.

**2. Report Failing Dies Count**  
**Function Name**: report\_failing\_dies\_count  
**ID**: 79\_5  
**Purpose**:  
Output the count of failing dies for the last tested wafer. This function focuses on reporting a single numerical metric that quantifies the yield performance from the most recent test.

**Signature**:

def report\_failing\_dies\_count(workflow\_id: int,

instruction\_id: int,

failing\_dies\_count: int) -> None:

"""

Output the count of failing dies for the last tested wafer.

Parameters:

workflow\_id (int): The workflow identifier.

instruction\_id (int): The instruction identifier.

failing\_dies\_count (int): The count of failing dies on the last tested wafer.

Returns:

None

"""

pass # Implementation goes here

**Used For**:

* **Workflow 79, Instruction 5**: Report Result – Output the count of failing dies for the last tested wafer.

This specification outlines the API function names, their purpose, parameters, and the list of instructions they support. Each function’s signature includes a clear docstring explaining the inputs and expected output, providing a solid basis for implementing the underlying logic in the semiconductor chip test data analytics domain.