Below is a proposed software specification for a set of Python API functions designed to execute the provided semiconductor chip test data analytics instructions. Each function is defined by its title (with numbering), a function name, a unique ID combining the workflow and instruction IDs, its purpose, a Python function header with a docstring, and the list of instructions for which the function is intended.

**Function 1: Compute Average Yield for a Lot**

**Function Name**: compute\_average\_yield  
**ID**: 47\_4  
**Purpose**:  
Calculates the average yield for all wafers within the most recent lot by aggregating individual wafer yields.

**Signature**:

def compute\_average\_yield(lot\_data: list) -> float:

"""

Compute the average yield for a given lot.

Parameters:

lot\_data (list): A list of yield values for individual wafers.

Returns:

float: The average yield of the lot.

"""

**Used For**:

* "Compute Average Yield – Aggregate the individual wafer yields and calculate the average yield for the lot."

**Function 2: Aggregate Voltage Metrics per Lot**

**Function Name**: aggregate\_voltage\_metrics  
**ID**: 6\_3  
**Purpose**:  
Aggregates voltage measurement data per lot by computing key statistical metrics (e.g., average, median, and standard deviation) for voltage readings.

**Signature**:

def aggregate\_voltage\_metrics(voltage\_data: list) -> dict:

"""

Aggregate voltage measurement data per lot by computing metrics such as average, median, and standard deviation.

Parameters:

voltage\_data (list): A list of voltage measurement values for a given lot.

Returns:

dict: A dictionary containing aggregated metrics (e.g., {'average': value, 'median': value, 'std\_dev': value}).

"""

**Used For**:

* "Aggregate Voltage Data - Compute aggregated metrics (e.g., average, median, standard deviation) for the voltage measurements per lot."

**Function 3: Aggregate E-test Metrics per Lot**

**Function Name**: aggregate\_etest\_metrics  
**ID**: 24\_3  
**Purpose**:  
Aggregates E-test measurement data for each lot by calculating the overall trend, typically via the average E-test value, to capture lot-level performance trends.

**Signature**:

def aggregate\_etest\_metrics(etest\_data: list) -> dict:

"""

Aggregate E-test measurement data for each lot to capture the overall trend (e.g., average E-test value per lot).

Parameters:

etest\_data (list): A list of E-test measurement values for a lot.

Returns:

dict: A dictionary containing the aggregated metric (e.g., {'average': value}).

"""

**Used For**:

* "Aggregate E-test Measurements: Compute aggregate metrics for each lot from the E-test data to capture the overall trend (e.g., average E-test value per lot)."

**Function 4: Compute Weekly Product Yield**

**Function Name**: compute\_weekly\_product\_yield  
**ID**: 42\_4  
**Purpose**:  
Calculates the average wafer-level yield for a specific product (e.g., product XYZ) on a weekly basis, aiding in trend analysis over time.

**Signature**:

def compute\_weekly\_product\_yield(yield\_data: list, weeks: list, product: str) -> dict:

"""

Compute the weekly average wafer-level yield for a specified product.

Parameters:

yield\_data (list): A list containing yield values for wafers.

weeks (list): A list of corresponding week identifiers.

product (str): The product identifier (e.g., 'XYZ').

Returns:

dict: A dictionary mapping each week to its average yield for the product.

"""

**Used For**:

* "Aggregate Averages: Compute the average wafer-level yield for product XYZ for each week."

**Function 5: Count Edge-Ring Failure Wafers per Week**

**Function Name**: count\_edge\_ring\_wafers  
**ID**: 65\_5  
**Purpose**:  
Counts the number of wafers that exhibit the edge-ring failure pattern for each week, which can help monitor failure trends over time.

**Signature**:

def count\_edge\_ring\_wafers(wafers\_data: list, weeks: list) -> dict:

"""

Count the total number of wafers exhibiting the edge-ring failure pattern per week.

Parameters:

wafers\_data (list): A list of wafer records with failure pattern information.

weeks (list): A list of corresponding week identifiers.

Returns:

dict: A dictionary mapping each week to the count of wafers with the edge-ring failure pattern.

"""

**Used For**:

* "Count Wafers per Week: Compute the total number of wafers exhibiting the edge-ring pattern for each week."

**Function 6: Assess Yield Recovery After Re-probing**

**Function Name**: assess\_yield\_recovery  
**ID**: 1\_4  
**Purpose**:  
Determines the yield recovery for each lot by computing the difference between the second probe yield and the first probe yield, thereby assessing the improvement after re-probing.

**Signature**:

def assess\_yield\_recovery(first\_probe\_yields: list, second\_probe\_yields: list) -> list:

"""

Assess the yield recovery for each lot by computing the differences between second probe yield and first probe yield.

Parameters:

first\_probe\_yields (list): A list of yield values from the first probe.

second\_probe\_yields (list): A list of yield values from the second probe.

Returns:

list: A list of yield recovery values for each lot.

"""

**Used For**:

* "Assess Yield Recovery - Determine the yield recovery by computing the differences between second probe yield and first probe yield for each lot."

**Function 7: Compute Correlation Between PCM Parameters and FT1 Yield**

**Function Name**: compute\_pcm\_ft1\_correlation  
**ID**: 13\_4  
**Purpose**:  
Performs a statistical correlation analysis between various PCM parameters and FT1 yield values using methods such as Pearson or Spearman coefficients.

**Signature**:

def compute\_pcm\_ft1\_correlation(pcm\_parameters: dict, ft1\_yields: list) -> dict:

"""

Compute statistical correlations between each PCM parameter and FT1 yield using Pearson or Spearman coefficients.

Parameters:

pcm\_parameters (dict): A dictionary where keys are PCM parameter names and values are lists of parameter measurements.

ft1\_yields (list): A list of FT1 yield values corresponding to the PCM parameter measurements.

Returns:

dict: A dictionary mapping each PCM parameter to its correlation metrics (e.g., {'pearson': value, 'spearman': value}).

"""

**Used For**:

* "Correlation Analysis - Compute statistical correlations between each PCM parameter and FT1 yield (using metrics like Pearson or Spearman coefficients)."

This specification outlines the API functions with clear definitions and consistent parameters that align with the requirements of semiconductor chip test data analytics.