Below is one possible software specification. In this design we have grouped similar instructions into API functions so that one function may support several instructions (as indicated in the “Used For” section by listing the Workflow/Instruction IDs and brief instruction summaries). You can later refine parameters or add extra options as needed.

1. **Title of the Function: Equipment Chain Consistency Analysis**  
   **Function Name**: analyze\_equipment\_chain\_consistency  
   **ID**: 2\_6, 2\_7, 2\_4  
   **Purpose**: Check whether consecutive lots with voltage failures are tested on the same equipment chain and correlate the chain consistency with the frequency of failures.  
   **Signature**:
2. def analyze\_equipment\_chain\_consistency(equipment\_data: pd.DataFrame, lot\_data: pd.DataFrame) -> dict:
3. """
4. Analyze the consistency of equipment chains across consecutive lots with voltage failures.
6. Parameters:
7. equipment\_data (pd.DataFrame): Data with equipment chain details (e.g., test house, tester, load board).
8. lot\_data (pd.DataFrame): Lot-level test data including voltage failure flags.
10. Returns:
11. dict: Results including chain consistency metrics, trend analysis, and potential hardware issue indicators.
12. """

**Used For**:

* + "Analyze Equipment Chain Consistency – Check if the same equipment chain appears consistently across consecutive lots with voltage failures." (Workflow 2, Instruction 6)
  + "Correlation and Trend Analysis – Perform trend analysis to correlate recurring equipment chain with voltage-related failure frequency." (Workflow 2, Instruction 7)
  + "Identify Consecutive Lots – Analyze the filtered data to determine consecutive lots with voltage-related failures." (Workflow 2, Instruction 4)

1. **Title of the Function: Voltage–Yield Correlation Analysis**  
   **Function Name**: analyze\_voltage\_yield\_correlation  
   **ID**: 6\_7, 6\_8  
   **Purpose**: Quantify and interpret the relationship between voltage measurement trends and lot-level yield performance.  
   **Signature**:
2. def analyze\_voltage\_yield\_correlation(voltage\_data: pd.DataFrame, yield\_data: pd.DataFrame) -> dict:
3. """
4. Perform statistical correlation analysis between voltage measurement trends and yield.
6. Parameters:
7. voltage\_data (pd.DataFrame): Voltage measurement data.
8. yield\_data (pd.DataFrame): Corresponding yield data for the lots.
10. Returns:
11. dict: Correlation coefficients, p-values, and interpretation of the relationship.
12. """

**Used For**:

* + "Perform Correlation Analysis – Use statistical methods (e.g., Pearson correlation) to quantify the relationship between voltage trends and yield." (Workflow 6, Instruction 7)
  + "Interpret Results – Analyze the correlation outputs to assess whether changes in voltage measurements correlate with yield trends." (Workflow 6, Instruction 8)

1. **Title of the Function: Soft Bin Yield Change Analysis**  
   **Function Name**: analyze\_soft\_bin\_yield\_change  
   **ID**: 18\_6  
   **Purpose**: Correlate changes in soft bin failure counts with yield data to determine if yield variations are driven by increased failures.  
   **Signature**:
2. def analyze\_soft\_bin\_yield\_change(soft\_bin\_data: pd.DataFrame, yield\_data: pd.DataFrame) -> dict:
3. """
4. Correlate soft bin failure counts with yield to assess if yield changes align with failure count variations.
6. Parameters:
7. soft\_bin\_data (pd.DataFrame): Data with soft bin failure counts.
8. yield\_data (pd.DataFrame): Lot-level yield data.
10. Returns:
11. dict: Correlation metrics and analysis results.
12. """

**Used For**:

* + "Analyze Yield Change – Correlate any changes in soft bin failure counts with yield data to see if yield decrease/increase aligns with increased failures." (Workflow 18, Instruction 6)

1. **Title of the Function: Tester and Load Board Impact Analysis**  
   **Function Name**: analyze\_tester\_loadboard\_impact  
   **ID**: 20\_6, 20\_8  
   **Purpose**: Compare yield statistics across different tester and load board combinations to identify trends and assess configuration impact.  
   **Signature**:
2. def analyze\_tester\_loadboard\_impact(yield\_data: pd.DataFrame, config\_data: pd.DataFrame) -> dict:
3. """
4. Compare yield performance across various tester and load board configurations and determine significant differences.
6. Parameters:
7. yield\_data (pd.DataFrame): Yield performance data.
8. config\_data (pd.DataFrame): Data containing tester and load board configuration details.
10. Returns:
11. dict: Comparative analysis report with statistical summaries and recommendations.
12. """

**Used For**:

* + "Perform Comparative Analysis – Compare yield statistics across different tester and load board combinations." (Workflow 20, Instruction 6)
  + "Draw Conclusions and Document Findings – Analyze visualizations and summaries to report on tester-specific load board impact on yield." (Workflow 20, Instruction 8)

1. **Title of the Function: E-test Trends Analysis**  
   **Function Name**: analyze\_etest\_trends  
   **ID**: 24\_6, 3\_4, 29\_7  
   **Purpose**: Plot time-series charts and perform trend analysis on E-test metrics and lot yields to identify fluctuations and patterns.  
   **Signature**:
2. def analyze\_etest\_trends(etest\_data: pd.DataFrame, yield\_data: pd.DataFrame) -> dict:
3. """
4. Analyze trends in E-test measurements and correlate them with lot-level yield performance.
6. Parameters:
7. etest\_data (pd.DataFrame): E-test measurement data.
8. yield\_data (pd.DataFrame): Yield data corresponding to the same time periods or lots.
10. Returns:
11. dict: Trend analysis results, including plots and statistical summaries.
12. """

**Used For**:

* + "Analyze Trends – Plot time-series charts for E-test metrics and lot yields." (Workflow 24, Instruction 6)
  + "Trend Analysis on E-test Measurements – Aggregate and analyze E-test data for fluctuations over time and within lots." (Workflow 3, Instruction 4)
  + "Review and Validate – Analyze the histogram of a selected E-test measurement to ensure correct distribution representation." (Workflow 29, Instruction 7)

1. **Title of the Function: Wafer Map Pattern Analysis**  
   **Function Name**: analyze\_wafer\_map\_patterns  
   **ID**: 25\_6, 65\_3, 18\_2, 37\_6, 21\_3  
   **Purpose**: Detect recurring wafer map patterns (such as edge-ring or grid) and assess their frequency and correlation with failing outcomes.  
   **Signature**:
2. def analyze\_wafer\_map\_patterns(wafer\_map\_data: pd.DataFrame) -> dict:
3. """
4. Analyze wafer map data to detect recurring patterns and evaluate their correlation with failure modes.
6. Parameters:
7. wafer\_map\_data (pd.DataFrame): Data containing wafer map details and failure locations.
9. Returns:
10. dict: Results including detected patterns, frequency metrics, and correlations with failures.
11. """

**Used For**:

* + "Correlation Analysis – Group and analyze frequency of detected patterns across probe card IDs and test programs." (Workflow 25, Instruction 6)
  + "Identify Edge-Ring Failure Pattern – Detect edge-ring patterns where failures concentrate along the wafer edge." (Workflow 65, Instruction 3 and Workflow 18, Instruction 2)
  + "Aggregate and Analyze Patterns – Cluster the detected patterns among failing wafers." (Workflow 37, Instruction 6)
  + "Identify Edge-Ring Failure Pattern – Check if a majority of failing dies are located along the wafer edge." (Workflow 21, Instruction 3)

1. **Title of the Function: Probe Touch Distribution Analysis**  
   **Function Name**: analyze\_probe\_touch\_distribution  
   **ID**: 37\_5, 37\_7  
   **Purpose**: Map probe touch coordinates and statistically analyze if specific touch locations are linked to increased yield loss.  
   **Signature**:
2. def analyze\_probe\_touch\_distribution(touch\_data: pd.DataFrame, yield\_data: pd.DataFrame) -> dict:
3. """
4. Map probe touch coordinates and analyze if certain touch locations correlate with higher yield loss.
6. Parameters:
7. touch\_data (pd.DataFrame): Data with probe touch coordinates.
8. yield\_data (pd.DataFrame): Corresponding wafer yield data.
10. Returns:
11. dict: Spatial analysis results with statistical significance of specific touch locations.
12. """

**Used For**:

* + "Analyze Probe Touch Distribution – Map probe touch coordinates and analyze their frequency on low-yield wafers." (Workflow 37, Instruction 5)
  + "Statistical Analysis – Assess whether yield loss is significantly higher when the probe touches at certain locations." (Workflow 37, Instruction 7)

1. **Title of the Function: Failure Pattern Classification**  
   **Function Name**: classify\_failure\_patterns  
   **ID**: 44\_5, 44\_3, 59\_4  
   **Purpose**: Classify the spatial distribution of failing dies to identify distinct patterns (e.g., region-specific clusters, edge-ring, center) and evaluate their persistence.  
   **Signature**:
2. def classify\_failure\_patterns(wafer\_data: pd.DataFrame) -> dict:
3. """
4. Classify failure patterns based on the spatial distribution of failing dies.
6. Parameters:
7. wafer\_data (pd.DataFrame): Data containing spatial coordinates of failing dies.
9. Returns:
10. dict: Classified patterns with metrics indicating persistence across lots.
11. """

**Used For**:

* + "Analyze Pattern Persistence Across Lots – Group wafers by lot and assess the recurrence of region-specific failure patterns." (Workflow 44, Instruction 5)
  + "Identify Concentrated Failure Patterns – Determine if failures are predominantly clustered in defined regions." (Workflow 44, Instruction 3)
  + "Classify Failure Patterns – Categorize spatial distributions based on pre-defined criteria." (Workflow 59, Instruction 4)

1. **Title of the Function: Statistical Significance Evaluation**  
   **Function Name**: evaluate\_statistical\_significance  
   **ID**: 78\_7  
   **Purpose**: Evaluate p-values and confidence intervals to determine the significance of predictive relationships (e.g., test bin failures predicting yield drops).  
   **Signature**:
2. def evaluate\_statistical\_significance(test\_results: dict) -> dict:
3. """
4. Evaluate statistical significance using p-values and confidence intervals.
6. Parameters:
7. test\_results (dict): Dictionary containing outputs from statistical tests.
9. Returns:
10. dict: Summary of significance evaluation with key metrics.
11. """

**Used For**:

* + "Evaluate Significance – Analyze p-values and confidence intervals to determine the significance of the predictive relationship." (Workflow 78, Instruction 7)

1. **Title of the Function: Test Bin and Program Analysis**  
   **Function Name**: analyze\_test\_bin\_and\_program  
   **ID**: 31\_4, 31\_5  
   **Purpose**: Extract and aggregate failure data by test bins and map failing dies to test programs for flagged lots.  
   **Signature**:
2. def analyze\_test\_bin\_and\_program(failure\_data: pd.DataFrame, test\_program\_data: pd.DataFrame) -> dict:
3. """
4. Analyze test bin failure data and map failing dies to test programs to determine which bins/programs contribute most to yield drops.
6. Parameters:
7. failure\_data (pd.DataFrame): Data containing failure counts and test bin identifiers.
8. test\_program\_data (pd.DataFrame): Data linking dies to test program configurations.
10. Returns:
11. dict: Analysis report with rankings of test bins and test program impact.
12. """

**Used For**:

* + "Test Bin Analysis – Extract and aggregate failure data by test bins for flagged lots." (Workflow 31, Instruction 4)
  + "Test Program Analysis – Map failing dies to test programs and rank their failure contributions." (Workflow 31, Instruction 5)

1. **Title of the Function: Failure Rate Analysis by Probing Cycles**  
   **Function Name**: analyze\_failure\_rate\_by\_cycle  
   **ID**: 25\_6  
   **Purpose**: Compare failure rates between early and late probing cycles using statistical tests to detect significant differences.  
   **Signature**:
2. def analyze\_failure\_rate\_by\_cycle(failure\_data: pd.DataFrame, cycle\_info: pd.DataFrame) -> dict:
3. """
4. Compare failure rates between early and late probing cycles using statistical methods.
6. Parameters:
7. failure\_data (pd.DataFrame): Data containing failure counts.
8. cycle\_info (pd.DataFrame): Information on probing cycle timing.
10. Returns:
11. dict: Results of statistical comparisons between early and late cycles.
12. """

**Used For**:

* + "Statistical Analysis – Compare failure rates between early and late probing cycles." (Workflow 25, Instruction 6)

1. **Title of the Function: Consistent Underperformers Filtering**  
   **Function Name**: filter\_consistent\_underperformers  
   **ID**: 9\_5  
   **Purpose**: Identify wafers with consistently low yield (e.g., yield below 95% in two or more consecutive weeks).  
   **Signature**:
2. def filter\_consistent\_underperformers(weekly\_yield\_data: pd.DataFrame, threshold: float = 95.0) -> pd.DataFrame:
3. """
4. Filter wafers that repeatedly show yield below the specified threshold over multiple weeks.
6. Parameters:
7. weekly\_yield\_data (pd.DataFrame): Weekly yield percentages for wafers.
8. threshold (float): Yield threshold (default 95%).
10. Returns:
11. pd.DataFrame: DataFrame containing wafers meeting the underperformance criteria.
12. """

**Used For**:

* + "Identify Consistent Underperformers – List wafers with a consistent yield below 95% over multiple weeks." (Workflow 9, Instruction 5)

1. **Title of the Function: Die Failure Cluster Analysis**  
   **Function Name**: analyze\_die\_failure\_clusters  
   **ID**: 67\_6  
   **Purpose**: Compute characteristics (size, density, position) of die failure clusters to understand their spatial distribution.  
   **Signature**:
2. def analyze\_die\_failure\_clusters(cluster\_data: pd.DataFrame) -> dict:
3. """
4. Compute characteristics of die failure clusters (e.g., size, density, position) to support correlation with process adjustments.
6. Parameters:
7. cluster\_data (pd.DataFrame): Data containing clustered failure information.
9. Returns:
10. dict: Analysis results summarizing cluster characteristics.
11. """

**Used For**:

* + "Analyze Cluster Characteristics – Compute metrics for each failure cluster." (Workflow 67, Instruction 6)

1. **Title of the Function: Spatial Failure Pattern Analysis**  
   **Function Name**: analyze\_spatial\_failure\_patterns  
   **ID**: 68\_4, 68\_6  
   **Purpose**: Apply spatial clustering to failing die coordinates and analyze clusters along with process adjustment data to infer potential process issues.  
   **Signature**:
2. def analyze\_spatial\_failure\_patterns(die\_coordinates: pd.DataFrame, process\_data: pd.DataFrame) -> dict:
3. """
4. Apply spatial clustering to identify patterns in failing die coordinates and correlate these with manufacturing process adjustments.
6. Parameters:
7. die\_coordinates (pd.DataFrame): Coordinates of failing dies.
8. process\_data (pd.DataFrame): Manufacturing process adjustments or logs.
10. Returns:
11. dict: Spatial analysis results with inferred process issues.
12. """

**Used For**:

* + "Spatial Analysis – Apply a clustering algorithm to failing die coordinates to identify spatial patterns." (Workflow 68, Instruction 4)
  + "Identify Root Causes – Analyze cluster characteristics with process adjustments to infer anomalies." (Workflow 68, Instruction 6)

1. **Title of the Function: Environmental Impact Analysis**  
   **Function Name**: analyze\_environmental\_impact  
   **ID**: 46\_5  
   **Purpose**: Evaluate the correlation between the bin 25 failure rate and environmental temperature readings during testing.  
   **Signature**:
2. def analyze\_environmental\_impact(failure\_data: pd.DataFrame, temperature\_data: pd.DataFrame) -> dict:
3. """
4. Analyze the relationship between bin 25 failure rates and environmental temperature during testing.
6. Parameters:
7. failure\_data (pd.DataFrame): Data with bin failure rates.
8. temperature\_data (pd.DataFrame): Environmental temperature readings.
10. Returns:
11. dict: Analysis results with correlation metrics and impact assessment.
12. """

**Used For**:

* + "Perform Correlation Analysis – Analyze the relationship between bin 25 failure rate and environmental temperature." (Workflow 46, Instruction 5)

1. **Title of the Function: Test Bin Classification Comparison**  
   **Function Name**: compare\_test\_bin\_classifications  
   **ID**: 1\_6  
   **Purpose**: Compare test bin distributions between first and second probe attempts to determine if classification changes are significant.  
   **Signature**:
2. def compare\_test\_bin\_classifications(first\_probe\_data: pd.DataFrame, second\_probe\_data: pd.DataFrame) -> dict:
3. """
4. Compare test bin distributions between first and second probe attempts using statistical tests.
6. Parameters:
7. first\_probe\_data (pd.DataFrame): Data from the first probing.
8. second\_probe\_data (pd.DataFrame): Data from the second probing.
10. Returns:
11. dict: Comparison results including statistical significance.
12. """

**Used For**:

* + "Compare Test Bin Classifications – Analyze changes in test bin distributions between probe attempts." (Workflow 1, Instruction 6)

1. **Title of the Function: Test Measurement Comparison Across Yields**  
   **Function Name**: compare\_test\_measurements\_across\_yield  
   **ID**: 30\_5  
   **Purpose**: Compare failing test measurements in low-yield wafers with those in normal-yield wafers to assess possible masking by bin limits.  
   **Signature**:
2. def compare\_test\_measurements\_across\_yield(low\_yield\_data: pd.DataFrame, normal\_yield\_data: pd.DataFrame) -> dict:
3. """
4. Compare test measurements between low-yield and normal-yield wafers to identify common failure signatures.
6. Parameters:
7. low\_yield\_data (pd.DataFrame): Failing test measurements for low-yield wafers.
8. normal\_yield\_data (pd.DataFrame): Test measurement data for normal-yield wafers.
10. Returns:
11. dict: Comparison analysis results.
12. """

**Used For**:

* + "Compare Test Measurements – Analyze whether failing test measurements in low-yield wafers are present in normal-yield wafers." (Workflow 30, Instruction 5)

1. **Title of the Function: Variance–Yield Correlation Analysis**  
   **Function Name**: correlate\_measurement\_variance\_with\_yield  
   **ID**: 29\_6  
   **Purpose**: Determine if higher variance in test measurements correlates with lower yield using statistical tests.  
   **Signature**:
2. def correlate\_measurement\_variance\_with\_yield(variance\_data: pd.DataFrame, yield\_data: pd.DataFrame) -> dict:
3. """
4. Analyze the correlation between variance in test measurements and wafer yield.
6. Parameters:
7. variance\_data (pd.DataFrame): Data with variance metrics for test measurements.
8. yield\_data (pd.DataFrame): Corresponding yield data.
10. Returns:
11. dict: Correlation results including statistical metrics.
12. """

**Used For**:

* + "Correlate Variance with Yield – Determine if wafers with higher test measurement variance correspond to lower yield." (Workflow 29, Instruction 6)

1. **Title of the Function: Correlate Yield Drops with E-test and Equipment Usage**  
   **Function Name**: correlate\_yield\_drops\_with\_etest\_and\_equipment  
   **ID**: 42\_8  
   **Purpose**: Analyze the relationship between sharp yield drops, changes in E-test site statistics, and equipment usage logs.  
   **Signature**:
2. def correlate\_yield\_drops\_with\_etest\_and\_equipment(yield\_data: pd.DataFrame, etest\_stats: pd.DataFrame, equipment\_logs: pd.DataFrame) -> dict:
3. """
4. Correlate sharp yield drops with changes in E-test site statistics and equipment usage.
6. Parameters:
7. yield\_data (pd.DataFrame): Yield trends.
8. etest\_stats (pd.DataFrame): E-test site statistics.
9. equipment\_logs (pd.DataFrame): Equipment usage logs.
11. Returns:
12. dict: Correlation analysis results with potential causes.
13. """

**Used For**:

* + "Correlate Data – Analyze the relationship between yield drops, E-test site statistics, and equipment usage logs." (Workflow 42, Instruction 8)

1. **Title of the Function: Failing Outcomes Filtering**  
   **Function Name**: filter\_failing\_outcomes  
   **ID**: 11\_3  
   **Purpose**: Filter test measurement records to isolate those that resulted in failing outcomes.  
   **Signature**:
2. def filter\_failing\_outcomes(test\_records: pd.DataFrame, failure\_criteria: dict) -> pd.DataFrame:
3. """
4. Filter test measurement records to identify failing outcomes based on defined criteria.
6. Parameters:
7. test\_records (pd.DataFrame): Data containing test measurement records.
8. failure\_criteria (dict): Criteria for what constitutes a failing outcome.
10. Returns:
11. pd.DataFrame: Filtered records that did not meet the test limits.
12. """

**Used For**:

* + "Filter Failing Outcomes – Identify test measurement records with failing outcomes." (Workflow 11, Instruction 3)

1. **Title of the Function: Reporting of Tester–Load Board Analysis**  
   **Function Name**: report\_tester\_loadboard\_analysis  
   **ID**: 20\_8  
   **Purpose**: Interpret and compile the analysis results (visualizations, statistical summaries) of tester and load board configurations into a detailed report with recommendations.  
   **Signature**:
2. def report\_tester\_loadboard\_analysis(analysis\_results: dict) -> str:
3. """
4. Compile a report from tester and load board analysis including methodology, key findings, and recommendations.
6. Parameters:
7. analysis\_results (dict): Results from comparative and correlation analyses.
9. Returns:
10. str: A comprehensive analysis report.
11. """

**Used For**:

* + "Draw Conclusions and Document Findings – Summarize visualizations and statistical summaries regarding tester-specific load board configurations." (Workflow 20, Instruction 8)

1. **Title of the Function: E-test Correlation Findings Interpretation**  
   **Function Name**: interpret\_etest\_correlation\_findings  
   **ID**: 49\_8  
   **Purpose**: Interpret correlation coefficients and visual trends from E-test data to derive process implications.  
   **Signature**:
2. def interpret\_etest\_correlation\_findings(correlation\_results: dict) -> dict:
3. """
4. Interpret correlation coefficients and visual trends from E-test measurements to determine process implications.
6. Parameters:
7. correlation\_results (dict): Results from E-test correlation analyses.
9. Returns:
10. dict: A summary of insights and potential process implications.
11. """

**Used For**:

* + "Interpret Findings – Analyze correlation coefficients and visual trends to document process implications." (Workflow 49, Instruction 8)

1. **Title of the Function: Significant Usage Changes Identification**  
   **Function Name**: identify\_significant\_changes\_in\_usage  
   **ID**: 27\_7  
   **Purpose**: Determine whether changes in prober or load board usage coincide with increases in failing logical tests.  
   **Signature**:
2. def identify\_significant\_changes\_in\_usage(usage\_data: pd.DataFrame, failure\_data: pd.DataFrame) -> dict:
3. """
4. Identify significant changes in prober or load board usage that correlate with increased failing logical tests.
6. Parameters:
7. usage\_data (pd.DataFrame): Data on prober/load board usage.
8. failure\_data (pd.DataFrame): Data on failing logical test outcomes.
10. Returns:
11. dict: Analysis results highlighting significant usage changes.
12. """

**Used For**:

* + "Identify and Highlight Significant Changes – Determine if increases in failing logical tests align with changes in prober or load board usage." (Workflow 27, Instruction 7)

1. **Title of the Function: Worst-Performing Wafer Identification**  
   **Function Name**: identify\_worst\_performing\_wafer  
   **ID**: 38\_2  
   **Purpose**: Identify, for each lot, the wafer with the lowest yield (i.e., worst performing).  
   **Signature**:
2. def identify\_worst\_performing\_wafer(wafer\_data: pd.DataFrame) -> pd.DataFrame:
3. """
4. Identify the worst-performing wafer in each lot based on the lowest yield values.
6. Parameters:
7. wafer\_data (pd.DataFrame): Wafer-level yield measurements.
9. Returns:
10. pd.DataFrame: Subset of wafer data identifying the worst-performing wafer per lot.
11. """

**Used For**:

* + "Identify Worst-Performing Wafer – Determine which wafer in each lot has the lowest yield." (Workflow 38, Instruction 2)

1. **Title of the Function: Yield Trend Analysis**  
   **Function Name**: analyze\_yield\_trend  
   **ID**: 39\_4  
   **Purpose**: Aggregate yield data over time or lots to identify overall trends and fluctuations in yield performance.  
   **Signature**:
2. def analyze\_yield\_trend(yield\_data: pd.DataFrame) -> dict:
3. """
4. Aggregate and analyze yield data over time or across lots to identify trends and fluctuations.
6. Parameters:
7. yield\_data (pd.DataFrame): Yield percentages for various lots and time periods.
9. Returns:
10. dict: Trend analysis results including key statistical metrics.
11. """

**Used For**:

* + "Yield Trend Analysis – Aggregate yield data to identify trends and fluctuations in yield performance." (Workflow 39, Instruction 4)

1. **Title of the Function: Yield Dips Identification**  
   **Function Name**: identify\_yield\_dips  
   **ID**: 2\_3  
   **Purpose**: Identify lots whose yield falls below a target threshold over the past two months.  
   **Signature**:
2. def identify\_yield\_dips(lot\_yield\_data: pd.DataFrame, target\_threshold: float) -> pd.DataFrame:
3. """
4. Identify lots with yield below the target threshold.
6. Parameters:
7. lot\_yield\_data (pd.DataFrame): Lot-level yield data.
8. target\_threshold (float): The yield threshold below which lots are flagged.
10. Returns:
11. pd.DataFrame: Filtered data of lots with yield below the threshold.
12. """

**Used For**:

* + "Identify Yield Dips – Flag lots that fall below the target yield threshold." (Workflow 2, Instruction 3)

1. **Title of the Function: Common Soft Bins Identification**  
   **Function Name**: identify\_common\_soft\_bins  
   **ID**: 6\_2  
   **Purpose**: Determine which soft bins appear most frequently in lots with wafer-level yield below 80%.  
   **Signature**:
2. def identify\_common\_soft\_bins(soft\_bin\_data: pd.DataFrame) -> dict:
3. """
4. Analyze soft bin data to compute frequency distributions and identify the most common soft bins.
6. Parameters:
7. soft\_bin\_data (pd.DataFrame): Data containing soft bin failure information.
9. Returns:
10. dict: Frequency distribution and common soft bins identified.
11. """

**Used For**:

* + "Identify Common Soft Bins – Analyze soft bin data for lots with low wafer-level yield." (Workflow 6, Instruction 2)

1. **Title of the Function: Logical Test Contribution Analysis**  
   **Function Name**: analyze\_logical\_test\_contributions  
   **ID**: 28\_6, 28\_7  
   **Purpose**: Calculate the contribution of each logical test to soft bin counts and use statistical checks to determine disproportionate contributions.  
   **Signature**:
2. def analyze\_logical\_test\_contributions(test\_data: pd.DataFrame) -> dict:
3. """
4. Calculate contribution percentages for logical tests and perform proportion tests to identify disproportionate effects.
6. Parameters:
7. test\_data (pd.DataFrame): Data with logical test outcomes and soft bin counts.
9. Returns:
10. dict: Analysis results including contribution metrics and statistical test outcomes.
11. """

**Used For**:

* + "Analyze Disproportionate Contributions – Calculate contribution percentages for logical tests." (Workflow 28, Instruction 6)
  + "Statistical Analysis – Perform proportion tests on logical test contributions." (Workflow 28, Instruction 7)

1. **Title of the Function: PCM Anomalies Analysis**  
   **Function Name**: analyze\_pcm\_anomalies  
   **ID**: 13\_7  
   **Purpose**: Assess whether anomalies in PCM test locations correlate with low FT1 yield and suggest adjustments related to foundry tuning.  
   **Signature**:
2. def analyze\_pcm\_anomalies(pcm\_data: pd.DataFrame, yield\_data: pd.DataFrame) -> dict:
3. """
4. Analyze PCM test location anomalies and correlate them with FT1 yield to recommend process adjustments.
6. Parameters:
7. pcm\_data (pd.DataFrame): Data containing PCM test location parameters.
8. yield\_data (pd.DataFrame): FT1 yield data.
10. Returns:
11. dict: Analysis results including correlation metrics and recommendations.
12. """

**Used For**:

* + "Spatial and Process Analysis – Assess if anomalies in PCM test locations contribute to low yield and are linked to foundry tuning issues." (Workflow 13, Instruction 7)

1. **Title of the Function: Comparative Yield Analysis Across Test Houses**  
   **Function Name**: analyze\_yield\_variation\_across\_test\_houses  
   **ID**: 7\_5  
   **Purpose**: Compare wafer-level yield performance across different test houses using statistical tests.  
   **Signature**:
2. def analyze\_yield\_variation\_across\_test\_houses(yield\_data: pd.DataFrame, test\_house\_info: pd.DataFrame) -> dict:
3. """
4. Apply statistical tests to compare yield performance across different test houses.
6. Parameters:
7. yield\_data (pd.DataFrame): Yield data for wafers.
8. test\_house\_info (pd.DataFrame): Information on test house identifiers.
10. Returns:
11. dict: Statistical comparison results and significance metrics.
12. """

**Used For**:

* + "Comparative Statistical Analysis – Determine if there are significant yield differences across test houses." (Workflow 7, Instruction 5)

1. **Title of the Function: Grid-Like Failure Pattern Correlation Analysis**  
   **Function Name**: analyze\_grid\_like\_failure\_correlation  
   **ID**: 10\_7  
   **Purpose**: Analyze the frequency of grid-like wafer map patterns in relation to specific tester–prober combinations and associated test data.  
   **Signature**:
2. def analyze\_grid\_like\_failure\_correlation(wafer\_map\_data: pd.DataFrame, configuration\_data: pd.DataFrame) -> dict:
3. """
4. Correlate grid-like wafer map failure patterns with tester–prober configurations and associated test bin/measurement data.
6. Parameters:
7. wafer\_map\_data (pd.DataFrame): Data with wafer map failure patterns.
8. configuration\_data (pd.DataFrame): Tester–prober configuration details.
10. Returns:
11. dict: Correlation analysis results.
12. """

**Used For**:

* + "Correlation Analysis – Analyze the frequency of grid-like patterns with tester–prober combinations." (Workflow 10, Instruction 7)

1. **Title of the Function: Dominant Failure Bin Determination**  
   **Function Name**: determine\_dominant\_failure\_bin  
   **ID**: 12\_3  
   **Purpose**: Identify the dominant failure bin for each wafer, which indicates the primary failure mode.  
   **Signature**:
2. def determine\_dominant\_failure\_bin(failure\_bin\_data: pd.DataFrame) -> dict:
3. """
4. Determine the dominant failure bin for each wafer based on failure bin counts.
6. Parameters:
7. failure\_bin\_data (pd.DataFrame): Data containing failure bin counts.
9. Returns:
10. dict: Dominant failure bin for each wafer.
11. """

**Used For**:

* + "Determine Dominant Failure Bin – Extract the dominant failure bin indicating the main failure mode." (Workflow 12, Instruction 3)

1. **Title of the Function: Test Bin–Tester Correlation Reporting**  
   **Function Name**: report\_test\_bin\_tester\_correlation  
   **ID**: 20\_8  
   **Purpose**: Interpret and report on the correlation between test bin distributions and tester configurations, including recommendations based on the findings.  
   **Signature**:
2. def report\_test\_bin\_tester\_correlation(correlation\_results: dict) -> str:
3. """
4. Interpret correlation results between test bin distributions and tester configurations and compile a detailed report.
6. Parameters:
7. correlation\_results (dict): Results from correlation analysis.
9. Returns:
10. str: A detailed report summarizing key findings and recommendations.
11. """

**Used For**:

* + "Interpretation and Reporting – Analyze correlation results and prepare a report on significant relationships." (Workflow 20, Instruction 8)

This specification outlines 33 API functions grouped by analysis type. Each function’s signature lists the expected input types (using, for example, Pandas DataFrames for tabular data) and returns a dictionary or DataFrame with the analysis results (or a report in the case of reporting functions). Adjust parameters or combine functions further as needed based on implementation details and additional requirements.