USING JMP TO AUTOMATE CAPACITY MODEL INPUT DERIVATION

IN THE SEMICONDUCTOR MANUFACTURING INDUSTRY

USING JMP TO AUTOMATE CAPACITY MODEL INPUT DERIVATION

IN THE SEMICONDUCTOR MANUFACTURING INDUSTRY

PROBLEM STATEMENT

A capacity model is an invaluable tool for driving manufacturing decisions, and its accuracy is directly affected by its inputs. As an Industrial Engineer at ON Semiconductor, one of my job functions was to derive and maintain the accuracy of my fab area inputs (Photolithography, Etch, Implant, Probe). The different tool processing types (batch/cluster/single piece) in photo, implant, etch, and probe varied greatly and amounted to over 8,000 unique tool and recipe level model inputs. I relied on tool logs and time studies to validate setup time (STIME) and processing time (PTIME).

Each tool vendor had their own method for logging tool events. The naming conventions varied which made parsing logic tool specific, and the method for obtaining log files frequently caused delay. In a high volume manufacturing environment where the mix is constantly changing, parsing enough tool log data to achieve a sufficient sample size for all the recipes was a difficult and manual process.

In addition to format variations, the processes were constantly evolving. For example, an SPC chart shift, a tool component failure, or a yield issue may require a process parameter adjustment. Common manufacturing questions arise: Do these changes impact the actual throughput of the tool? Does the new robot enhancement improve the speed at the bottleneck portion of the tool? Does tool A perform at the same speed as tool B, C and D?

Manual time studies and tool log parsing could not keep up with the demands of four wafer fab areas. An automated approach to data analysis was required.

WHY JMP?

I started with a familiar open source statistical package that I had brief exposure to in my graduate studies. I have never considered myself an advanced coder, and I struggled to find time to devote to learning new syntax with the continued demands of the fab.

I continued my search for software solutions, and found JMP to be the clear stand out among the various packages available. The approachable scripting language, intuitive user interface, and extensive online resources allowed me, a first-time JMP user, to bring my data automation ideas to fruition in a short period of time.

Understanding very little about the JMP Scripting Language, I was able to leverage the GUI and export the resultant scripts. I used these generated script snippets to learn the syntax as I went, making progress along the way. The time saved from automating even small portions of the process was realized within the first week, and allowed for more time to build upon these scripts.

SOLUTION

Ultimately, I created several JSL scripts to save our company time and money. A simple 5 step approach was used:

1. Connect to an Oracle database
2. Extract and sort the data required for calculations
3. Perform calculations
4. Screen the data for outliers
5. Generate summary tables and reports

There are different script types (PTIME, STIME, etc.) depending on the desired output metric, and flavors within those (single wafer, batch, etc.) to align with how the tool processes wafers:

Process Time (PTIME)	**Setup Time (STIME)**	**Miscellaneous**
Single Wafer	Single Wafer	Recipe transitions delays
Single Wafer Chamber (cluster)	Single Wafer Chamber (cluster)	Implant beam tuning
Batch	Batch	

Table 1: Script Types and Flavors

An ASML i-Line (photolithography tool) processes wafers one at a time, in series. I would run the following scripts:

- Single Wafer PTIME – produces average process time per unit at a tool and recipe level
- Single Wafer STIME – produces average setup time per lot at a tool and recipe level
- Recipe Transition Delays – produces raw transition delay data that can be formulated into a From-To Matrix to aid in dispatch logic

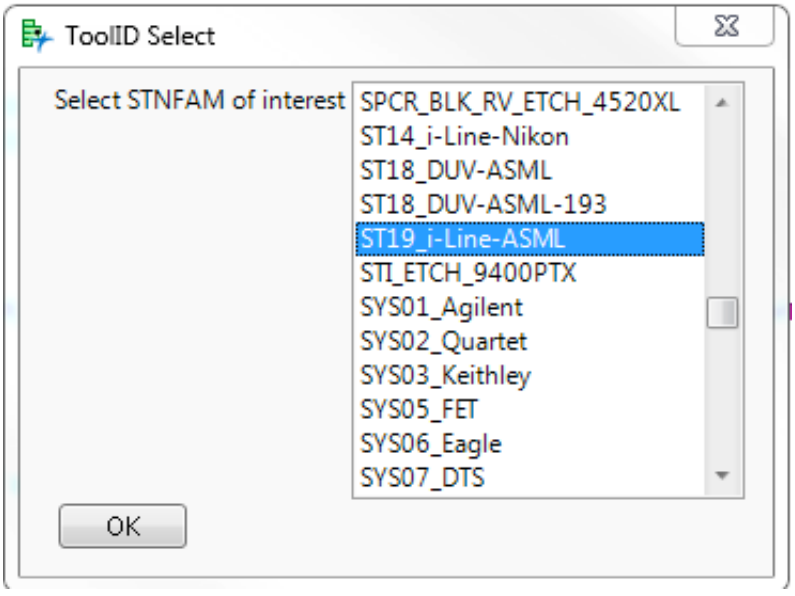
Initialize input variables:

- Start Date
- End Date

```
1  // ********************************************************************************
2  // ********************************************************************************
3  // Tool Matching and Outlier Analysis Script - Single Wafer PTIME
4  // Author: Jason Kintz
5  // ********************************************************************************
6  // ********************************************************************************
7
8
9
10
11 //initialize variables
12 //NOTE: '%' can be used as a wildcard in strings
13     startDate = "01-01-2015";  //date format= MM-DD-YYYY ; must be BEFORE endDate
14     endDate = "04-23-2015";    //date format= MM-DD-YYYY ; must be AFTER startDate
15     // buffer value specified in 'Outlier Removal' popup window acts as a USCL and
16     // LSCL buffer (in minutes) to ensure reasonable variation is captured:
17     //  uscl = upperq + (buffer+(1.5*(upperq-lowerq)));
18     //  lscl = lowerq - (buffer+(1.5*(upperq-lowerq)));
```

Run script

- A user dialog prompt appears, asking what Station Family (or tool group) you would like to return data from:



We group similar tool ID's into Station Families (STNFAMs) or ToolGroups:

Tool ID	**STNFAM**
TRK05603	ST19_i-Line-ASML
TRK05604	ST19_i-Line-ASML
TRK05605	ST19_i-Line-ASML
TRK05606	ST19_i-Line-ASML
TRK05607	ST19_i-Line-ASML

Table 2: ToolID to STNFAM mapping

This Tool – STNFAM mapping lives in an oracle table that the script references to prompt for the user's input. Based on the selection, I then store the ToolID's associated with that STNFAM to a list variable ("stn") used to build a dynamic query.

SQL query

For the Single Wafer PTIME script, I am only extracting 'Wafer_Complete' time stamp events from the database that occurred during the time range of interest and on the ToolID's of interest:

```
"SELECT * FROM dBTable

  WHERE (TOOL_ID IN ("||stn||"))
AND NEW_EVENT_ID LIKE 'Wafer_Complete'
AND (TIME_STAMP BETWEEN TO_DATE('"||startDate||"','MM-DD-YYYY') AND
TO_DATE('"||endDate||"','MM-DD-YYYY'))

  ORDER BY
TOOL_ID, RUN_ID, TIME_STAMP"
```

RUN_ID is a unique identifier for each pass through a tool (assigned at the lot or batch level). The SQL query handles the data sorting with the ORDER BY clause (sorts by TOOL_ID, then RUN_ID, then TIME_STAMP).

Mean Time Between Wafers (MTBW) Calculation

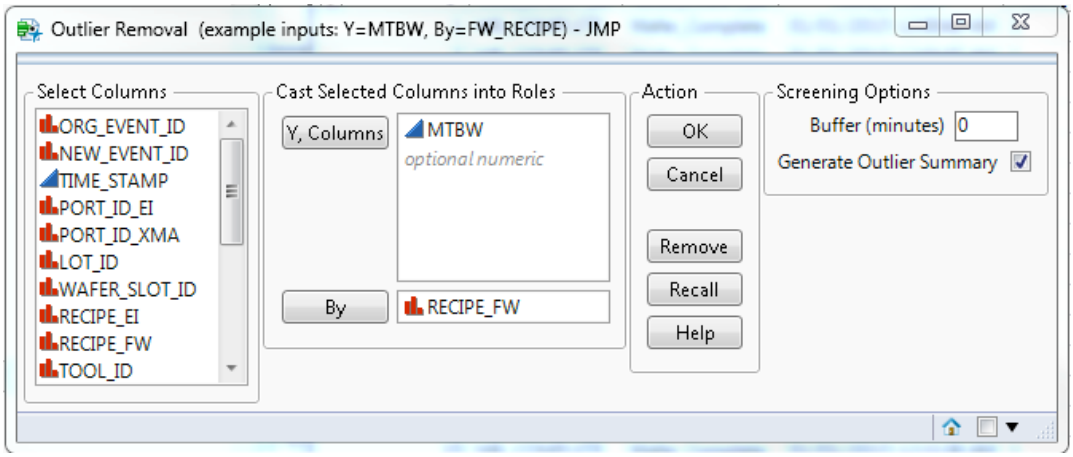
With the sorting already completed, the data table imported into JMP is ready for a simple Time Between Wafer Complete Time Stamps calculation (current row – previous row):

We exclude the first wafer of each RUN_ID (with an ELSE clause) to segregate Setup Time from the dataset. This also avoids erroneous data points due to lot and recipe transition. At this point, each row in the "MTBW" column actually represents actual time between wafers (in minutes). We will ultimately take the average of these data points (hence the name Mean Time Between Wafers).

Outlier screening logic

Now that the data is extracted, and we have a calculated Time Between Wafer completes, we need to screen for extreme outliers. We do not want to include datapoints in our average that are the result of a tool alarm, maintenance event, or other non-standard activity.

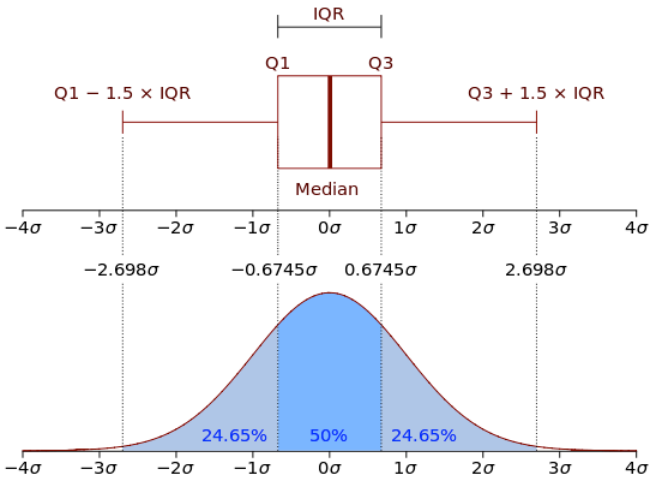
We need to tell the script what data columns to use as inputs for determining the Upper and Lower Screening Limits, and how to partition the dataset. A user input dialog box appears:



- The **'Y, Columns'** input is our calculated metric (time between wafer complete time stamps).
- The **'By'** input tells the script how to partition this large data table.
    - In this case, we are slicing by Recipe.

So for each recipe, we are essentially plotting the observed MTBW values, then establishing an upper and lower threshold. Everything outside of this acceptable range gets flagged (and removed) as an outlier.

Upper and Lower Screening limits - Tukey's Boxplot Method

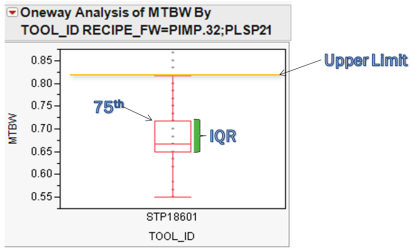


This provides a quick way to flag and remove extreme values from the dataset before we take an average of the Time Between Wafer Complete data points. While Tukey's method is considered an 'informal' test for identifying outliers (and should be used with caution), for our purposes, it does a sufficient job of screening suspect values from the base dataset.

A buffer variable (user defined via Outlier Screening dialog box) can be used to expand the screening limit range. This was a request from Process Engineering (and a modification to Tukey's method) that is only utilized on tools that are very consistent (tight IQR lead to flagging outliers that were just outside original limits).

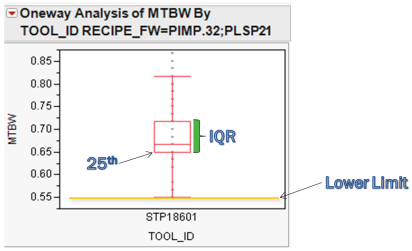
Upper Screening Limit

= 75th Quartile + [(1.5*IQR)+buffer]



Lower Screening Limit

= 25th Quartile - [(1.5*IQR)+buffer]



This portion of the logic was built off of Ian Cox's "Data Filtering" script in the JMP User Community File Exchange. The outlier screening dialog box from his script was used in my adaptation and he even provided help via email as I hit roadblocks trying to apply similar looping logic. Another reason that I chose JMP for this automation project was the strong online community and extensive library of example scripts I could use as learning tools.

The new version of JMP (version 12) now includes an 'Explore Outliers Utility' found under the Modeling Utilities section. This new feature offers four different options to identify and explore outliers:

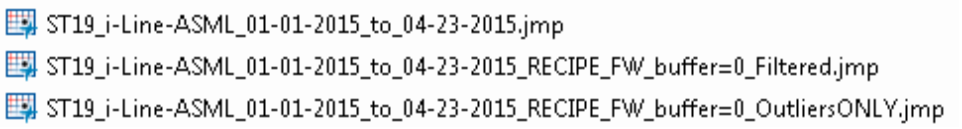
1. Quantile Range Outliers
2. Robust Fit Outliers
3. Multivariate Robust Outliers
4. Multivariate k-Nearest Neighbor Outliers

I look forward to exploring these new features as they have the potential to improve and streamline the current scripts.

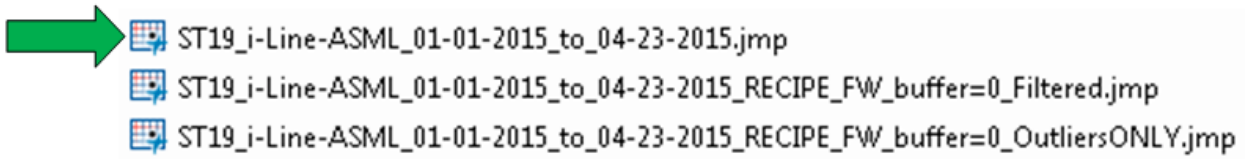
Output Tables and Reports

Ultimately, three main JMP data tables are exported. Script variables and user inputs are used in file naming, which helps organization and end user interpretation:

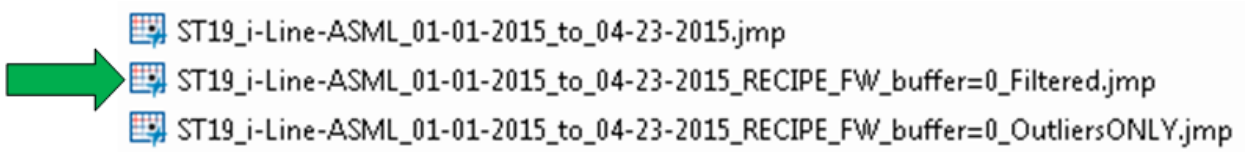
STNFAM_StartDate_to_EndDate_ByVariable_BufferVariable



The first data table generated is the base (original) dataset. This is the original SQL query output with an additional column for the MTBW calculation.

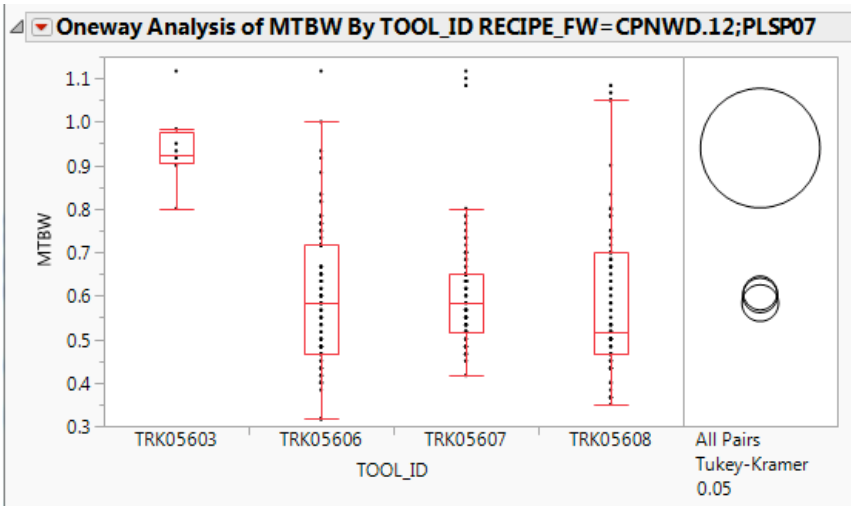


The second data table is the output after outlier screening has occurred (referred to as the Filtered dataset). The user input buffer variable (0 minutes in this case) is displayed in the file name. In this table, all of the outlier values (points outside the Upper and Lower limits) were screened and replaced with missing values (represented by dots in the cell):

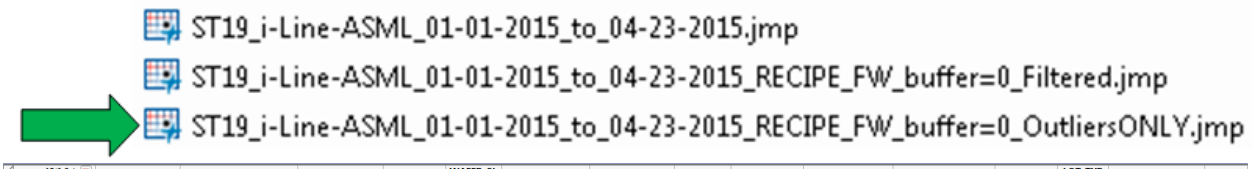


	NEW_EVENT_ID	TIME_STAMP	PORT_ID_XMA	LOT_ID	WAFER_SLOT_ID	RECIPE_EI	RECIPE_FW	TOOL_ID	TECHNOLOGY	PLAN	STEP	DEVICE	LOT_TYPE	RUN_ID	MTBW
274	Wafer_Complete	01/07/2015 7:51:50 AM	TRK05603-IDX A	GAQ47104.1	wA01	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
275	Wafer_Complete	01/07/2015 7:52:12 AM	TRK05603-IDX A	GAQ47104.1	wA02	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.37
276	Wafer_Complete	01/07/2015 7:52:47 AM	TRK05603-IDX A	GAQ47104.1	wA03	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.58
277	Wafer_Complete	01/07/2015 7:53:08 AM	TRK05603-IDX A	GAQ47104.1	wA04	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.35
278	Wafer_Complete	01/07/2015 7:53:42 AM	TRK05603-IDX A	GAQ47104.1	wA05	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.57
279	Wafer_Complete	01/07/2015 7:54:50 AM	TRK05603-IDX A	GAQ47104.1	wA06	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
280	Wafer_Complete	01/07/2015 7:55:37 AM	TRK05603-IDX A	GAQ47104.1	wA07	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.78
281	Wafer_Complete	01/07/2015 7:56:07 AM	TRK05603-IDX A	GAQ47104.1	wA08	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.50
282	Wafer_Complete	01/07/2015 7:56:50 AM	TRK05603-IDX A	GAQ47104.1	wA09	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.72
283	Wafer_Complete	01/07/2015 7:58:02 AM	TRK05603-IDX A	GAQ47104.1	wA10	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
284	Wafer_Complete	01/07/2015 7:58:31 AM	TRK05603-IDX A	GAQ47104.1	wA11	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.48
285	Wafer_Complete	01/07/2015 7:59:15 AM	TRK05603-IDX A	GAQ47104.1	wA12	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.73
286	Wafer_Complete	01/07/2015 8:00:30 AM	TRK05603-IDX A	GAQ47104.1	wA13	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
287	Wafer_Complete	01/07/2015 8:01:01 AM	TRK05603-IDX A	GAQ47104.1	wA14	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.52
288	Wafer_Complete	01/07/2015 8:01:40 AM	TRK05603-IDX A	GAQ47104.1	wA15	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.65
289	Wafer_Complete	01/07/2015 8:03:09 AM	TRK05603-IDX A	GAQ47104.1	wA16	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
290	Wafer_Complete	01/07/2015 8:03:35 AM	TRK05603-IDX A	GAQ47104.1	wA17	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.43
291	Wafer_Complete	01/07/2015 8:04:22 AM	TRK05603-IDX A	GAQ47104.1	wA18	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.78
292	Wafer_Complete	01/07/2015 8:05:05 AM	TRK05603-IDX A	GAQ47104.1	wA19	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.72
293	Wafer_Complete	01/07/2015 8:05:34 AM	TRK05603-IDX A	GAQ47104.1	wA20	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.48
294	Wafer_Complete	01/07/2015 8:06:16 AM	TRK05603-IDX A	GAQ47104.1	wA21	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.70
295	Wafer_Complete	01/07/2015 8:06:45 AM	TRK05603-IDX A	GAQ47104.1	wA22	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.48
296	Wafer_Complete	01/07/2015 8:07:24 AM	TRK05603-IDX A	GAQ47104.1	wA23	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.65
297	Wafer_Complete	01/07/2015 8:08:29 AM	TRK05603-IDX A	GAQ47104.1	wA24	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
298	Wafer_Complete	01/07/2015 8:08:58 AM	TRK05603-IDX A	GAQ47104.1	wA25	Block.34.01;PLS...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	0.48

A tool matching report is generated for each Recipe to identify improvement opportunities:



The third data table is the inverse of the second (referred to as the Outlier dataset); whereas only the screened outlier values are shown (all other datums in the MTBW column are replaced with missing values).

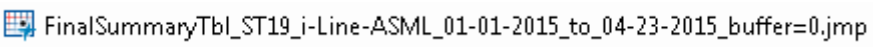


17/0 Cols

	NEW_EVENT_ID	TIME_STAMP	PORT_ID_XMA	LOT_ID	WAFER_SLOT_ID	RECIPE_EI	RECIPE_FW	TOOL_ID	TECHNOLOGY	PLAN	STEP	DEVICE	LOT_TYPE	RUN_ID	MTBW
274	Wafer_Complete	01/07/2015 7:51:50 AM	TRK05603-IDX A	GAQ471041	wA01	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
275	Wafer_Complete	01/07/2015 7:52:12 AM	TRK05603-IDX A	GAQ471041	wA02	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
276	Wafer_Complete	01/07/2015 7:52:47 AM	TRK05603-IDX A	GAQ471041	wA03	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
277	Wafer_Complete	01/07/2015 7:53:08 AM	TRK05603-IDX A	GAQ471041	wA04	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
278	Wafer_Complete	01/07/2015 7:53:42 AM	TRK05603-IDX A	GAQ471041	wA05	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
279	Wafer_Complete	01/07/2015 7:54:50 AM	TRK05603-IDX A	GAQ471041	wA06	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	1.13
280	Wafer_Complete	01/07/2015 7:55:37 AM	TRK05603-IDX A	GAQ471041	wA07	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
281	Wafer_Complete	01/07/2015 7:56:07 AM	TRK05603-IDX A	GAQ471041	wA08	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
282	Wafer_Complete	01/07/2015 7:56:50 AM	TRK05603-IDX A	GAQ471041	wA09	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
283	Wafer_Complete	01/07/2015 7:58:02 AM	TRK05603-IDX A	GAQ471041	wA10	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	1.20
284	Wafer_Complete	01/07/2015 7:58:31 AM	TRK05603-IDX A	GAQ471041	wA11	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
285	Wafer_Complete	01/07/2015 7:59:15 AM	TRK05603-IDX A	GAQ471041	wA12	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
286	Wafer_Complete	01/07/2015 8:00:30 AM	TRK05603-IDX A	GAQ471041	wA13	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	1.25
287	Wafer_Complete	01/07/2015 8:01:01 AM	TRK05603-IDX A	GAQ471041	wA14	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
288	Wafer_Complete	01/07/2015 8:01:40 AM	TRK05603-IDX A	GAQ471041	wA15	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
289	Wafer_Complete	01/07/2015 8:03:09 AM	TRK05603-IDX A	GAQ471041	wA16	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	1.48
290	Wafer_Complete	01/07/2015 8:03:35 AM	TRK05603-IDX A	GAQ471041	wA17	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
291	Wafer_Complete	01/07/2015 8:04:22 AM	TRK05603-IDX A	GAQ471041	wA18	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
292	Wafer_Complete	01/07/2015 8:05:05 AM	TRK05603-IDX A	GAQ471041	wA19	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
293	Wafer_Complete	01/07/2015 8:05:34 AM	TRK05603-IDX A	GAQ471041	wA20	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
294	Wafer_Complete	01/07/2015 8:06:16 AM	TRK05603-IDX A	GAQ471041	wA21	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
295	Wafer_Complete	01/07/2015 8:06:45 AM	TRK05603-IDX A	GAQ471041	wA22	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
296	Wafer_Complete	01/07/2015 8:07:24 AM	TRK05603-IDX A	GAQ471041	wA23	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.
297	Wafer_Complete	01/07/2015 8:08:29 AM	TRK05603-IDX A	GAQ471041	wA24	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	1.08
298	Wafer_Complete	01/07/2015 8:08:58 AM	TRK05603-IDX A	GAQ471041	wA25	Block.34.01;PLSP...	Block.34;PLSP12	TRK05603	ONC35EE	ONC35EE-2P-3LM	Sal-Block-Mask-CED	G0A01RF-...	PRD	A20150107071323	.

In addition to the 3 main data tables, two summary tables are generated. The final summary table presents the metric of interest:

- the Mean Time Between Wafer Completes (MTBW) at the Recipe resolution
- units = minutes per wafer



RECIPE_FW	N Rows	MTBW (w/ Outliers)	N (less 1st wfr)	25th Qtl	75th Qtl	interQtl Range	MTBW (Outliers Removed)	# of Outliers	% Outliers
P1screen.03;PLSP18	25680	0.92092	24588	0.61667	1.15000	0.53	0.89179	440	1.7
PSD.04;PLSP07	21435	0.65326	20558	0.48333	0.71667	0.23	0.59667	1281	6.0
NSD.05;PLSP07	21202	0.63410	20336	0.48333	0.71667	0.23	0.59588	1160	5.5
Block.34;PLSP12	20300	0.64404	19447	0.48333	0.71667	0.23	0.58998	1305	6.4
MET1.10;PLSP10	15641	0.87033	14985	0.56667	1.13333	0.57	0.86184	77	0.5
Met2.13;PLSP10	15295	1.03385	14653	0.78333	1.21667	0.43	0.99208	385	2.5
Isl.01;PLSP30	10911	0.80774	10447	0.55000	0.98333	0.43	0.77583	287	2.6
Poly.03;PLSP30	10906	0.84489	10450	0.55000	1.16667	0.62	0.83475	65	0.6
PFLD.46;PLSP62	10315	0.72936	9883	0.50000	0.90000	0.40	0.71295	93	0.9
VIA1.11;PNSP32	9692	1.12761	9277	0.86667	1.35000	0.48	1.09111	226	2.3

Along with the metric of interest (MTBW Outliers Removed), I display the total sample (N Rows), the MTBW before we screened any outliers (MTBW w/Outliers), the IQR (interQtl Range), and an outlier summary.

Lastly, a summary table by ToolID is generated:

SummaryTbl_byToolID_ST19_01-01-2015_to_04-23-2015_buffer=0_filtered.jmp

	RECIPE_FW	N Rows	Mean(MTBW, TRK05603)	Mean(MTBW, TRK05605)	Mean(MTBW, TRK05606)	Mean(MTBW, TRK05607)	Mean(MTBW, TRK05608)	N(MTBW, TRK05603)	N(MTBW, TRK05605)	N(MTBW, TRK05606)	N(MTBW, TRK05607)	N(MTBW, TRK05608)
2	PPHR.40;PLSP07	29073	0.61948	0.57871	0.57670	0.59761	0.60100	4375	5184	7403	3935	5327
3	Met1.10;PLSP77	27624	1.51412	•	•	1.51735	1.51082	6909	0	0	5508	6590
4	P1screen.03;PLSP18	25680	1.02514	0.86939	0.91892	0.90198	0.87831	1866	12980	2659	2506	4137
5	PSD.04;PLSP07	21435	0.62167	•	0.57360	0.60337	0.60000	3862	0	6087	3777	5551
6	NSD.05;PLSP07	21202	0.61818	•	0.57908	0.59710	0.59595	3814	0	5406	4426	5530
7	Block.34;PLSP12	20300	0.61661	0.57680	0.57130	0.59269	0.59824	2912	7515	820	3618	3277
8	MET1.10;PLSP10	15641	0.94515	0.82002	0.86452	0.86013	0.81247	2706	164	4218	3278	4542
9	Met2.13;PLSP10	15295	1.05423	0.93645	1.02345	0.98743	0.92236	3063	91	4107	2680	4327
10	Isl.01;PLSP30	10911	•	•	0.77852	0.77316	•	0	0	5066	5094	0
11	Poly.03;PLSP30	10906	•	•	0.84917	0.82589	•	0	0	3954	6431	0
12	PFLD.46;PLSP62	10315	•	•	0.73694	0.71218	0.68317	0	0	3885	2848	3057
13	VIA1.11;PNSP32	9692	1.12903	1.08562	1.13345	1.09754	1.02499	1062	4758	1133	1055	1043

A 3 month time frame for a STNFAM consisting of 4-6 tools usually completes in about 5 minutes (only limited by speed of the SQL query).

Other Scripts

The batch tool script runtime is significantly faster than single wafer due to the size of the resultant data tables (querying for batch start/complete instead of individual wafer level time stamps). I can pull 1 year of batch tool data for a STNFAM consisting of 4-6 tools in about 5 minutes. The STIME scripts see a similar benefit due to only querying lot level events.

The Recipe Transition Changeover script is the most complicated logic as we need to quantify the time between recipe transitions and determine if a track-in gap influenced the data point. We used the Recipe Transition Changeover script to build a From-To Delay Matrix to identify significant recipe changeover delays and further optimize our dispatching logic.

RESULTS

ON Semiconductor has been using the scripts at their Oregon fab for over a year, and realized a total savings of ~1 Industrial Engineer's time across the department. This process was selected as a "Global Industrial Engineering Initiative" for 2015 with plans to roll out the scripts to other manufacturing sites in Idaho, Belgium, Czech Republic, Japan, and Malaysia.