

# ToolWindow Analysis

Statistical analysis comparing user engagement between manually-opened and automatically-opened tool windows in IDEs.

## Project Overview

This little project analyzes user interaction patterns with tool windows to determine whether there's a significant difference in session duration between windows opened manually versus those opened automatically.

## Key Findings

### Statistical Summary

**Manual Opens (n=621):** - **Mean:** 13.54 minutes - **Median:** 0.20 minutes  
- **Q1:** 0.04 minutes | **Q3:** 2.25 minutes - **Std Dev:** 72.10 minutes (understandable, because values can range from 0 up to 24 hours exclusively)

**Auto Opens (n=986):** - **Mean:** 56.55 minutes - **Median:** 2.89 minutes - **Q1:** 0.56 minutes | **Q3:** 17.86 minutes - **Std Dev:** 191.45 minutes (understandable, because values can range from 0 up to 24 hours exclusively)

### Statistical Significance

- **Mann-Whitney U Test:**  $p < 0.001$  (highly significant)
- **Effect Size (Cohen's d):** -0.275 (small effect)
- **Bootstrap 95% CI for Median Difference:** [-3.40, -2.23] minutes
- **Conclusion:** Manual sessions are statistically significantly **shorter** than auto-opened sessions

## Visualizations

Code produces two visualization files:

1. **toolwindow\_analysis.png** - Four-panel comparison:
  - Distribution histograms
  - Box plot comparison
  - Cumulative distribution functions (CDFs)
  - Key statistical metrics
2. **toolwindow\_violin.png** - Violin plot showing distribution shapes (capped at 30 min for clarity)

## Project Structure

```
toolwindow-analysis/
    toolwindow_analysis.py
    toolwindow_data.csv
```

```
matched_sessions.csv  
toolwindow_analysis.png  
toolwindow_violin.png  
requirements.txt  
Dockerfile  
docker-compose.yml  
README.md
```

## Methodology

### 1. Data Processing Approach

**Assumptions:** - Events are ordered chronologically per user - Each “opened” event should eventually be followed by a “closed” event, otherwise they are not considered - Sessions longer than 24 hours (1,440 minutes) are considered outliers/errors - Timestamps are in milliseconds since epoch

**Handling Messy Data:** - **Orphaned Opens (235):** Open events without corresponding closes (likely due to crashes, incomplete logging, or ongoing sessions) - **Orphaned Closes (8):** Close events without preceding opens (data collection started mid-session) - **Outliers (23):** Sessions exceeding 24 hours were removed as unrealistic (outliers)

**Matching Strategy:** - Stack-based algorithm per user (LIFO - Last In, First Out) - Each “opened” event pushed to stack with timestamp and open\_type - Each “closed” event pops most recent open and calculates duration - Ensures proper nesting of overlapping sessions

### 2. Statistical Methods

**Descriptive Stats:** - Mean, median, standard deviation, and quartiles for both groups (manual, auto) - Distribution visualizations (histograms, box plots, CDFs)

**Hypothesis Testing:** - **Mann-Whitney U Test:** Non-parametric test chosen because distributions are heavily right-skewed (not normally distributed) - Tests null hypothesis: no difference in session duration between groups

**Effect Size:** - **Cohen’s d:** Standardized measure of difference between groups - Interpretation: Small (0.2), Medium (0.5), Large (0.8)

**Confidence Intervals:** - **Bootstrap resampling (10,000 iterations):** Provides good confidence intervals for median difference without assuming normality

### 3. Data Quality Metrics

- **Total Events:** 3,503
- **Matched Sessions:** 1,630

- **After Outlier Removal:** 1,607 sessions
- **Data Retention Rate:** 95.8% (1,607/1,677 potential sessions)

## Running the Code

### Using Docker (Recommended)

Run the following from project's repository terminal.

```
docker-compose up --build
```

## Dependencies

- `pandas` 1.3.0
- `numpy` 1.21.0
- `matplotlib` 3.5.0
- `seaborn` 0.11.0
- `scipy` 1.7.0

## Output Files

1. `matched_sessions.csv` - Clean dataset with matched open/close pairs
  - Columns: `user_id`, `duration_minutes`, `open_type`
2. `toolwindow_analysis.png` - Main visualizations
3. `toolwindow_violin.png` - Distribution comparison

## Interpretation & Recommendations

### Key Insights

1. **Auto-opened windows have significantly longer session durations** (median: 2.89 min vs 0.20 min)
2. **High variability in both groups** suggests diverse usage patterns
3. **Small effect size** indicates practical difference may be modest despite statistical significance

### Possible Explanations

- **Manual opens:** Users intentionally open tool when needed: quick, focused usage
- **Auto opens:** Windows open by themselves: longer sessions as users may leave them open while working

### Post Conclusion Recommendations

1. Consider auto-opening for features that benefit from longer, persistent visibility

2. For quick-reference tools, manual opening may be more appropriate
3. Implement smart auto-close for idle auto-opened windows to manage screen

## OutPut Summary

Data overview

Rows: 3,503

Matched sessions: 1,630

Orphaned opens: 235

Orphaned closes: 8

Number of removed outliers (>24 hours): 23

Manual opens, n=621:

Mean: 13.538 min

Median(Q2): 0.201 min

Std dev: 72.101 min

Q1: 0.036 min

Q3: 2.254 min

Auto opens, n=986:

Mean: 56.548 min

Median(Q2): 2.890 min

Std dev: 191.449 min

Q1: 0.557 min

Q3: 17.864 min

Mann-Whitney U test:

Test stat: 155366.5000

p-value: 0.000000

Result: HIGHLY significant (p < 0.001)

Cohen's d, effect size:

d = -0.275

Interpretation: SMALL effect

Bootstrap 95% confidence interval:

Median diff: -2.689 min

95% CI: [-3.395, -2.227] min

Manual sessions are SHORTER