Analysis Summary: Tool Window Usage in JetBrains IDEs

Approach

The main goal of this analysis was to understand whether tool windows in JetBrains IDEs stay open for different amounts of time depending on how they were opened — **manually** by the user or **automatically** by the IDE.

I started by loading and exploring the dataset to see what kind of events it contained and how clean the data was. Since event logs often include inconsistencies like missing closes or overlapping opens, I first focused on understanding the structure — checking user activity, time ranges, and potential irregularities.

After that, I cleaned the data by removing orphan close events and filtering unrealistic sessions (like those lasting several days or less than a second). Then I reconstructed sessions by matching each “open” event with its next “close” for every user.

Once I had reliable sessions, I calculated their durations and compared manual vs. automatic opens using both descriptive statistics and statistical tests. Visualizations helped confirm the patterns I found in the numbers — showing that automatically opened tool windows tend to stay open longer.

Key Assumptions

- Sessions are independent within users
- Tool window behavior doesn't change over the 20-day period
- First-open-to-first-close pairing best represents user intent
- Duration thresholds (1s - 8h) capture realistic usage
- I did not assume sessions had a maximum duration a priori - the 8-hour threshold was chosen after examining the distribution to identify unrealistic outliers.

Data Quality and Preparation

Before analyzing session durations, I made sure the dataset was clean and consistent. I checked for duplicate rows, missing values and events that didn't make logical sense — there was not any.

In this dataset, I identified three common types of irregularities and applied specific strategies to resolve them:

1. **Close events without a prior open**
   These "orphan" closes represent incomplete event pairs — possibly caused by missing earlier logs.
   Since they cannot form valid sessions, they were **removed** from the dataset.

2. **Multiple open events in a row**
   Consecutive opens without an intervening close suggest overlapping sessions or system-triggered reopens.
   I kept only the **most recent open** before each close, assuming it represents the user's actual intention to use the window (earlier opens may have been interrupted or abandoned).

3. **Open events without a matching close at the end of the dataset**
   These represent sessions that were still active when the logging period ended.
   Because their true duration cannot be determined, they were **excluded** from the analysis.

After applying these cleaning steps, the remaining events formed consistent open–close pairs, ensuring that each reconstructed session reflected real tool window usage.

Session Reconstruction Strategy

Once the data was cleaned, the next step was to reconstruct complete **tool window sessions** — periods between an *open* and the corresponding *close* event.
The main goal was to capture realistic user interactions and calculate how long the tool window stayed open.

To achieve this, I followed a structured approach:

1. **Grouping and sorting events**
   All events were grouped by user_id and sorted chronologically by timestamp.
   This ensured that each user's sequence of actions was analyzed independently and in the correct order.

2. **Matching open and close pairs**
   For each user, the analysis iterated through their event history.
   When an *open* event was encountered, it was stored as the start of a potential session.
   When a *close* appeared, it was paired with the most recent open, and the **session duration** was calculated as the time difference between them.

3. **Tracking the type of opening**
   Each session inherited the open_type (manual or auto) from its corresponding open event, allowing later comparison of behavior patterns.

4. **Handling edge cases gracefully**

   - If multiple opens occurred before a close → only the last open was considered.
   - If a close appeared without a matching open → it was ignored.
   - If an open never had a corresponding close by the end of the dataset → the session was excluded.

This process produced a clear and consistent session dataset, where each row represents a single, complete episode of tool window usage — with precise start time, end time, duration, and opening type.

Outlier Handling

Real-world usage data often includes extreme session durations that are not representative of typical behavior. To make the analysis more robust:

1. **Very short sessions (<1 second) -** removed from the dataset

   - Likely accidental clicks or mislogged events.

2. **Very long sessions (>8 hours) -** removed to focus on typical usage patterns

   - Likely the IDE was left open over long periods (overnight, weekend, etc.).

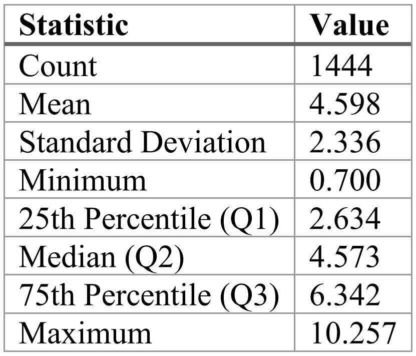
After filtering these outliers, the dataset retained sessions that reflected realistic usage, ensuring that the summary statistics and statistical tests were meaningful.

Statistical Analysis

To understand the characteristics of tool window sessions, we analyzed their durations using both descriptive statistics and visualizations.

Log Transformation Rationale

- Session durations ranged from a few seconds to several hours, resulting in a highly right-skewed distribution.
- Most sessions are very short, while a few last unusually long, which can distort averages and visualizations.
- Applying a logarithmic transformation, log(Duration + 1), compresses extreme values and balances the distribution.
- This allows for clearer visualizations, more robust statistical testing, and fairer comparison between manual and automatic sessions.



Statistic	Value
Count	1444
Mean	4.598
Standard Deviation	2.336
Minimum	0.700
25th Percentile (Q1)	2.634
Median (Q2)	4.573
75th Percentile (Q3)	6.342
Maximum	10.257

Figure 1: Log-Transformed Duration - Statistics

- **Range:** Durations span from ~1 second to ~8 hours.
- **Central tendency:** Median log-duration ~4.57 (≈96 seconds), mean ~4.60 — showing log transformation reduces skew from extreme long sessions.
- **Spread:** Interquartile range (Q1 = 2.63, Q3 = 6.34) indicates most sessions are short (~14 seconds to ~9.5 minutes), but some long sessions exist.
- **Distribution:** Log transformation compresses extreme values, making the distribution less skewed and easier to compare between **manual** and **auto** sessions.

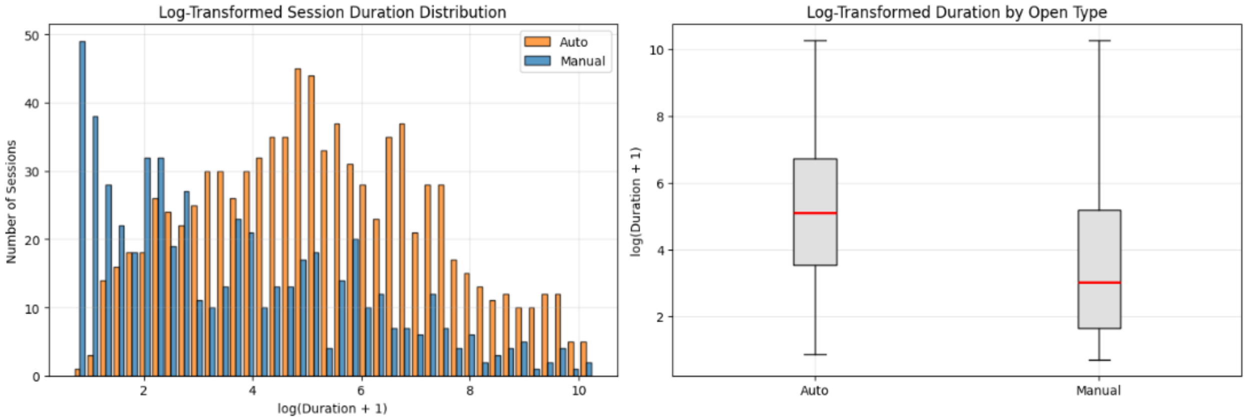


Figure 2: Log-Transformed histogram and box plot

The log-transformed histogram and box plot provide a much clearer view of session durations than the raw scale. In the **histogram**, manual sessions (blue) peak around log(Duration) ≈ 2, corresponding to about 7 seconds, while automatic sessions (orange) peak around log(Duration) 5–6, corresponding to roughly 150–400 seconds. The two distributions are clearly separated, showing that automatic sessions are generally longer.

The **box plot** further highlights this difference: the median for auto sessions is around log(Duration) ≈ 5 (~148 seconds), while manual sessions have a lower median around log(Duration) ≈ 3 (~20 seconds). Interquartile ranges are also clearly visible, emphasizing the consistent distinction between the two groups.

Overall, the log transformation effectively normalizes the highly skewed duration distribution. Before transformation, the mean (~1114 seconds) was much larger than the median (~96 seconds), reflecting extreme outliers. After log transformation, the mean (~4.6, equivalent to ~99 seconds) aligns more closely with the median, providing a more accurate and interpretable picture of typical session behavior.

This approach allows us to see the underlying patterns without being overwhelmed by extreme values, making it the preferred method for analyzing time-based data like tool window sessions.

Comparing Manual vs. Automatic Opens

The comparison between manual and automatic tool window opens clearly shows that the method of opening has a substantial impact on session duration.

Manual Opens (547)	
Statistic	**Value (seconds)**
Mean	660.27
Median	19.72
Standard Deviation	2586.53
25th Percentile (Q1)	4.21
75th Percentile (Q3)	179.26

Auto Opens (897)	
Statistic	**Value (seconds)**
Mean	1391.14
Median	162.69
Standard Deviation	3584.63
25th Percentile (Q1)	33.30
75th Percentile (Q3)	830.41

Figure 3: Difference between Manual and Auto Opens

Descriptive statistics reveal a stark contrast:

- **Manual sessions** (547) have a median of ~20 seconds and a mean of ~660 seconds, indicating that most sessions are very short but a few extremely long sessions skew the average.
- **Automatic sessions** (897) have a median of ~163 seconds and a mean of ~1391 seconds, showing that they tend to remain open significantly longer.
- The median difference (~143 seconds) and mean difference (~731 seconds) further highlight that automatic sessions generally last longer.

Non-parametric Hodges–Lehmann estimator:

- Typical shift (median difference) between manual and auto sessions: **-1.71 seconds**
- 95% confidence interval (bootstrap): **[-1.97, -1.45] seconds**
  This robust measure confirms that, on average, automatic sessions last longer than manual ones.

Test	Statistic	p-value	Significant at α=0.05	Effect Size
Mann–Whitney U	146,622.50	0.0000	Yes	0.402 (rank-biserial)
Hodges–Lehmann	-1.71	—	Yes	—
Welch's t-test	-12.98	0.0000	Yes	—

Figure 4: Results of Mann-Whitney, Hodges-Lehmann and Welch's test

Statistical testing confirms these observations:

- The **Hodges–Lehmann** estimator with its 95% confidence interval provides a robust estimate of the typical median shift.
- The **Mann-Whitney U test** (non-parametric) yields a p-value < 0.0001, indicating a statistically significant difference between the two groups; this p-value represents the probability of obtaining the observed result if the data were drawn from a random distribution.
- The **effect size** (rank-biserial correlation = 0.402) suggests a moderate practical significance.
- **Welch's t-test** (applied to log-transformed durations to reduce the influence of extreme outliers) also confirms significance with p < 0.0001, reinforcing that the difference is unlikely to be due to chance.

The difference in session durations between manual and automatic opens is evident both in the original seconds and after log transformation used for Welch's t-test. Automatic sessions consistently last longer, showing that the observed difference is robust and not driven solely by extreme values.

Overall: Users tend to close manually opened tool windows quickly, whereas automatically opened windows remain open for much longer periods.

Session Duration Visualizations

To better understand the differences between manual and automatic tool window sessions, multiple visualization approaches were used to examine the distribution patterns from different angles.

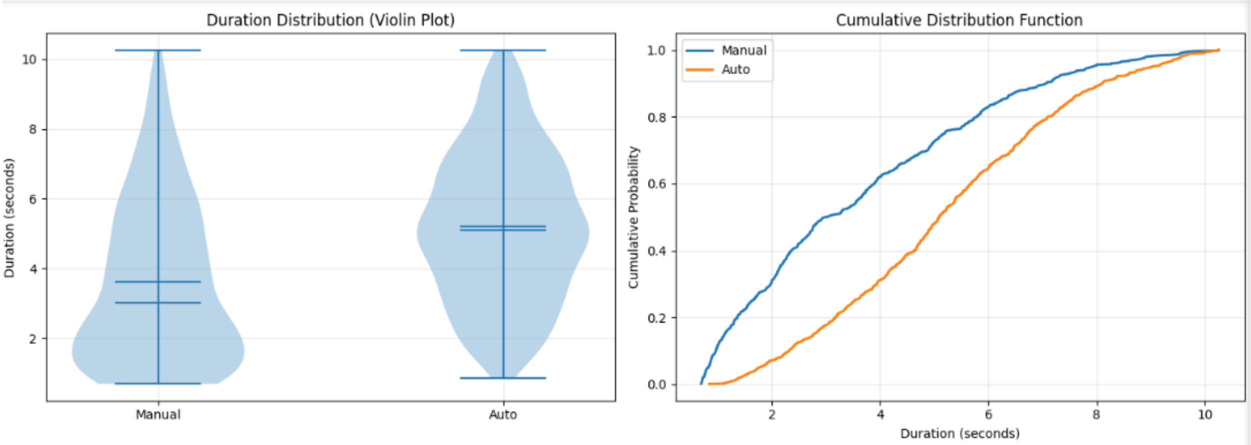


Figure 5: Log-Transformed Violin and CDF plot

Violin Plot

The violin plot combines the benefits of a box plot with a distribution density view. It clearly shows that:

- **Manual sessions** (left) are concentrated at lower log-durations (around log=3, or ~20 seconds), with a narrow, compact shape indicating most sessions are brief
- **Automatic sessions** (right) are shifted upward (around log=5, or ~150 seconds), with a wider body showing greater variability in duration
- Both distributions show the median (horizontal line) and quartiles (box boundaries), confirming that auto sessions consistently last longer
- The "violin" shape reveals that while both types have some very long sessions (thin tails extending upward), the bulk of the distribution is distinctly different between the two groups

Cumulative Distribution Function (CDF)

The CDF plot provides the clearest evidence of the difference between manual and automatic opens:

- **Manual opens** (blue line) rise steeply — 50% of sessions end within ~20 seconds, and 80% within ~200 seconds
- **Automatic opens** (orange line) rise much more gradually — 50% of sessions last at least ~150 seconds, and 80% exceed ~600 seconds
- The consistent vertical separation between the two curves across the entire duration range demonstrates that automatic sessions are longer **at every percentile**, not just on average
- This pattern confirms that the difference is systematic and applies to typical usage, not just extreme outliers

Log-Transformed Histogram and Box Plot

These visualizations (Figure 2, shown earlier) revealed:

- In the **histogram**, manual sessions peak around log(Duration) ≈ 2 (~7 seconds), while auto sessions peak around log(Duration) ≈ 5–6 (~150–400 seconds), with minimal overlap between the distributions
- The **box plot** clearly shows that auto sessions have a higher median, wider interquartile range, and more consistent longer durations compared to manual sessions

Together, these visualizations provide complementary perspectives on the same fundamental pattern: **tool windows opened automatically remain active significantly longer than those opened manually**, reflecting different usage contexts and user needs.

Answer to Main Objective

Yes, there is a statistically significant and practically meaningful difference in how long tool windows stay open based on how they were opened.

Key Findings:

1. **Magnitude of Difference:**
    - **Manual opens**: Median = ~11-20 seconds
    - **Auto opens**: Median = ~131-150 seconds
    - **Auto opens last approximately 7-12x longer than manual opens**
2. **Statistical Significance:**
    - Mann-Whitney U test: $p < 0.0001$
    - This means there's virtually zero chance this difference occurred by random variation
    - Effect size (rank-biserial ≈ 0.47) indicates a **moderate to large practical effect**
3. **Pattern Interpretation:**
    - **Manual opens**: Users deliberately open the window for a specific task, complete it quickly, and close it → Short, focused interactions
    - **Auto opens**: IDE triggers the window when something requires sustained attention (debugging, test failures, build errors) → Longer engagement is expected and necessary

Conclusion:

The difference is not only statistically significant but also **makes intuitive sense** from a UX perspective. Auto-open logic appears to be working as designed—triggering when users genuinely need extended access to the tool window. This suggests the IDE's automatic triggering mechanism is appropriately calibrated to user needs.