Individual Assignment: Analyzing Performance Data

Tássyla Lissa Lima

September 19, 2025

Abstract

This report presents an analysis of CPU usage from four versions of a software system. Using statistical techniques and data visualization, this assignment investigates the data distribution, the existence of statistically significant differences between versions, the magnitude of these differences, and the evolution of performance over time. The goal is to draw evidence-based conclusions about the impact of the different versions on the system's CPU consumption.

1 Analysis and Results

All analyses in this report were performed using Python. The source code, raw data files, and generated figures are available in a GitHub repository.[1]

1.1 Normality Analysis of the Data

Analysis: To decide whether the CPU usage data for each software version follows a normal distribution, both visual and formal statistical methods were used.

Visual inspection was conducted using histograms (Figure 1) and Quantile-Quantile (Q-Q) plots (Figure 2). The histograms reveal that the distributions appear to be right-skewed, with a high frequency of lower CPU values and a long tail towards higher values. The Q-Q plots provide further visual evidence against normality, showing that the data points consistently and significantly deviate from the theoretical diagonal line at the tails.

For a formal statistical assessment, the Shapiro-Wilk test was applied to each dataset. The resulting p-values were all $p < 0.0001$.

Conclusion: For all four versions, the p-value obtained from the Shapiro-Wilk test is substantially lower than the significance level of $\alpha = 0.05$. This statistical result, strongly supported by visual evidence from both histograms and Q-Q plots, leads to a definitive rejection of the null hypothesis of normality. It is concluded that the performance data for all tested versions are **not normally distributed**. This finding requires the use of non-parametric statistical tests for subsequent comparative analyses in this report.

1.2 Statistical Difference between base_version and version_1

Analysis: To determine if there is a statistically significant difference between the CPU performance of `base_version` and `version_1`, a hypothesis test was conducted. Based on the conclusion from the previous question that the data is not normally distributed, the non-parametric **Mann-Whitney U test** was selected. The null hypothesis (H_0) for this test states that there is no difference between the distributions of CPU usage for the two versions.

The test was performed on the "TOTAL_CPU" data from both versions, yielding a U-statistic of 148007.5 and a p-value of $p < 0.0001$.

[1] `https://github.com/tassyla/log6309e-assignment-1`

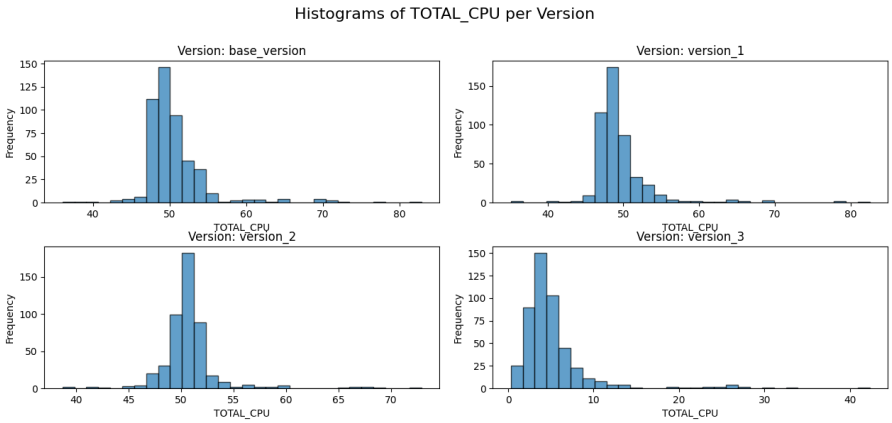

Figure 1: Histograms of TOTAL_CPU usage for each software version, showing right-skewed distributions.

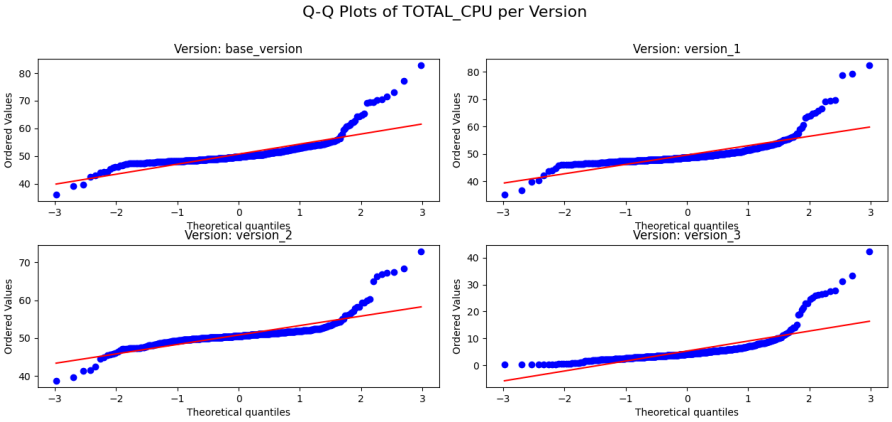

Figure 2: Q-Q plots of TOTAL_CPU usage for each software version. The significant deviation of the data points from the red line indicates non-normality.

Conclusion: The resulting p-value is substantially lower than the significance level of $\alpha = 0.05$. Therefore, we **reject** the null hypothesis. This provides evidence that there **is a statistically significant difference** in CPU consumption between `base_version` and `version_1`.

1.3 Magnitude of the Difference

Analysis: While the previous result confirmed a statistically significant difference, it did not quantify its magnitude. To measure the practical importance of the performance change between `base_version` and `version_1`, the effect size was calculated using **Cohen's d**.

The analysis yielded a Cohen's d value of -0.2464. The mean CPU usage for the `base_version` was 50.67, while for `version_1` it was 49.61. The negative sign of Cohen's d indicates a reduction in the mean CPU usage from the baseline to version 1.

Conclusion: A Cohen's d of -0.2464 has an absolute value that is conventionally interpreted as a **small** effect size. This result indicates that the change in CPU consumption from `base_version` to `version_1` is not only statistically significant but also represents a practically small, yet beneficial, improvement in performance.

1.4 Temporal Analysis of base_version

Analysis: To investigate whether the performance of the `base_version` changed over the monitoring period, a time series analysis was conducted. A line plot was generated to visually inspect the trend, and Spearman's rank correlation was used for a formal statistical test.

The line plot (Figure 3) shows considerable volatility in CPU usage but does not reveal an obvious upward or downward trend. For a quantitative measure, the Spearman correlation test was performed, yielding a correlation coefficient (ρ) of **-0.0909** with a p-value of **0.0462**.

Conclusion: The p-value of 0.0462 is marginally below the significance level of $\alpha = 0.05$, leading us to reject the null hypothesis. This indicates that there is a **statistically significant** downward trend. However, the Spearman correlation coefficient is very close to zero, indicating that the magnitude of this trend is **very weak**. Therefore, the conclusion is that while a subtle downward trend in CPU usage can be statistically detected, its practical effect is likely negligible.

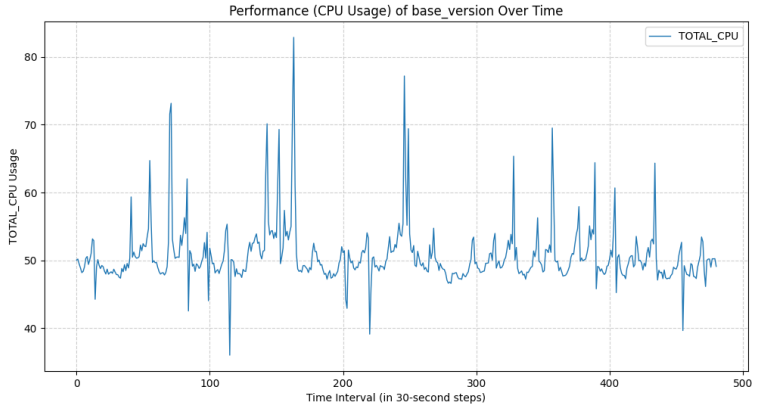

Figure 3: CPU usage of `base_version` over the monitoring period.

1.5 Grouping and Ranking the Versions

Analysis: To compare CPU performance across all four software versions, the Kruskal-Wallis H-test was performed, followed by Dunn's post-hoc test. The Kruskal-Wallis test resulted in a highly significant p-value ($p < 0.0001$), confirming that significant performance differences exist among the versions.

Dunn's post-hoc test with Bonferroni correction was then used to identify the specific pairs of versions that differ. The boxplot in Figure 4 visually confirms the substantial differences, and Table 1 summarizes the p-values from Dunn's test.

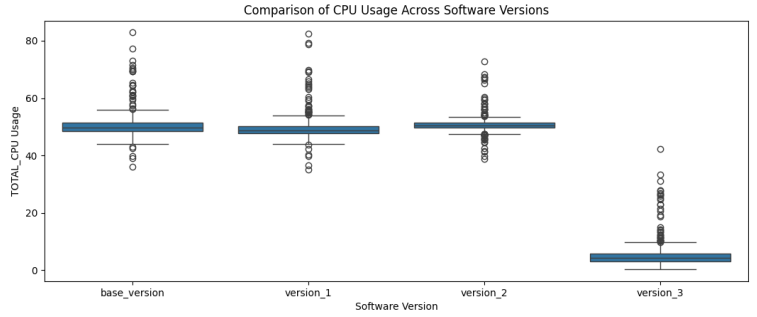

Figure 4: Boxplots comparing the distribution of CPU usage across all four software versions.

Table 1: P-values from Dunn's Post-Hoc Test for Pairwise Comparisons

Version	`base_version`	`version_1`	`version_2`
`version_1`	< 0.0001	-	-
`version_2`	< 0.001	< 0.0001	-
`version_3`	< 0.0001	< 0.0001	< 0.0001

All p-values are less than 0.05, indicating a statistically significant difference in every pairwise comparison.

Conclusion: The results from Dunn's test show that all pairwise comparisons yielded p-values substantially less than 0.05. This indicates that **every version is statistically different from every other version**. Therefore, each version forms its own distinct performance group. Based on the median CPU usage for each version, the final performance ranking from highest to lowest CPU consumption is as follows: (1) `version_2` (Median CPU: 50.53); (2) `base_version` (Median CPU: 49.66); (3) `version_1` (Median CPU: 48.72); (4) `version_3` (Median CPU: 4.22)

2 Discussion and Conclusions

The statistical analysis of the CPU performance data gave clear insights into the impact of each software version.

The main conclusions are:

- **Statistical Significance vs. Practical Impact:** The statistical tests confirmed that **every software version has a statistically distinct CPU performance profile**. However, the performance of `version_3` represents a change that is both statistically and practically significant, with a drastic reduction in CPU usage. The differences between the other versions vary by less than 2 percentage points.
- **Performance Evolution:** The performance progression was not linear. `Version 1` showed a **small but statistically significant performance improvement** (lower CPU usage) compared to the `base_version`. However, this gain was reversed in `Version 2`, which consumed more CPU than `version_1` and `base_version`.
- **Radical Optimization in Version 3:** `Version 3` stands out dramatically. It demonstrated a **massive reduction in CPU usage**. This is a fundamental shift in system behavior. Such a change could result from a major architectural optimization or the fix of a major bug.
- **Baseline Stability:** The `base_version` exhibited **stable performance** over time. Although a statistical test detected a mathematically significant downward trend, the trend's strength was so weak as to be **practically negligible**. This indicates the testing environment and workload were stable, providing a reliable baseline for comparison.

In summary, this quantitative analysis provides invaluable feedback for the development team. The regression in `version 2`, although small, should be noted. The dramatic improvement in `version 3` is a major success but must be understood to ensure the CPU reduction did not introduce unintended side effects or remove essential functionality. These findings exemplify how data analysis in the DevOps pipeline can lead to informed decisions and better overall software quality.