UNIVERSITÀ DEGLI STUDI DI GENOVA

DIBRIS

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY,
BIOENGINEERING, ROBOTICS AND SYSTEM ENGINEERING

RESEARCH TRACK 2

Third Assignment

Statistical Analysis

Author:

Delala Hocine

Student ID:

s5445123

Professors:

Carmine Tommaso Recchiuto

June 2, 2023

Contents

1 Introduction

Statistical analysis is a powerful tool for evaluating the performance of algorithms in various domains. It encompasses a wide range of methods, including descriptive statistics to summarize and describe data, inferential statistics to make inferences or predictions about a population based on sample data, hypothesis testing to evaluate the significance of relationships or differences, regression analysis to model relationships between variables, and many other techniques.

In this report, I present a statistical analysis of two different algorithm implementations for the same task (Research Track1 first assignment), one developed by colleague Baba Hadj Said and the other is my implementation . The goal is to perform a statistical analysis by performing different tests to test the hypothesis which will be presented later.

2 Experiment Description

Data was collected by running both code implementations in different environments, resulting in multiple performance measurements for each implementation. The collected data was then subjected to statistical analysis to draw meaningful conclusions.

2.1 Used Tools

I worked with the code(algorithm) of the first assignment of RT1, focusing and comparing two implementations:

- My implementation :`https://github.com/hocinedl/ResearchTrack1_FirstAssignment`.
- Baba's implementation:`https://github.com/babahadjsaid/Research-Track-1-Assignment-1`.

The objective of RT1 first assignment was to create a program that search for silver tokens in the environment (arena) and place it close to the golden tokens creating pairs of silver and golden tokens.

In order to carry out a good and reliable statistical analysis, I modified the code such that I can retrieve some parameters (like execution time). I used also Jupyter notebook for developing graphs that were helpful in the analysis.

2.2 Arena Configuration

In order to make different configurations of the placement of the silver tokens I modified the code to make the tokens randomly placed and I variate the number of tokens. At the end I ended up with the following Arena Configurations :

- 4 tokens
- 5 tokens
- 6 tokens
- 8 tokens

The position of the tokens is random and it is not in a circle shape as it was in the standard arena, meaning that each time the tokens were added in random positions. The configurations used are shown bellow :

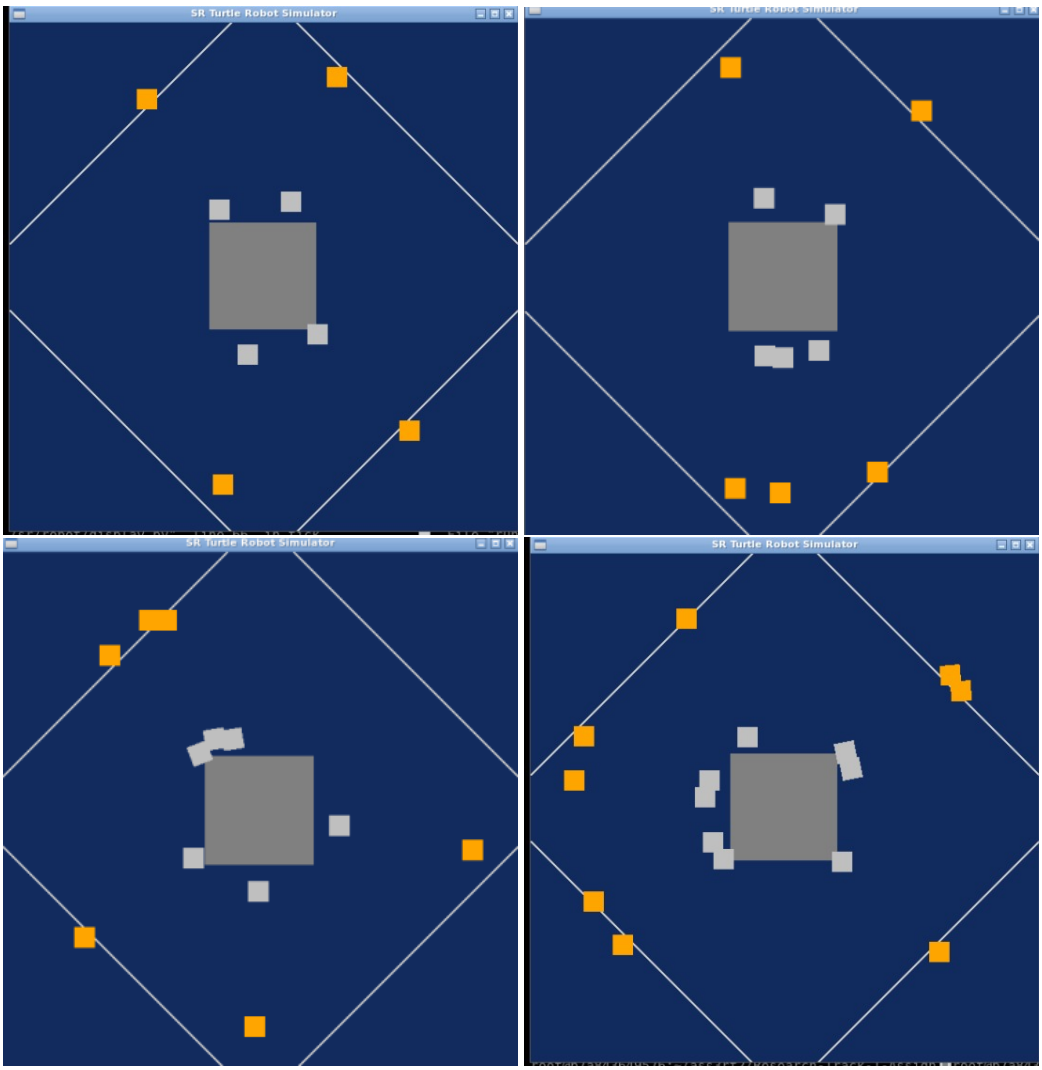

Figure 1: Different Environment Configurations (Arena)

2.3 Data Collection

For each code I performed the following experiments:

- For all the configurations (4,5,,6,8 tokens) I performed 10 attempts for both codes (my code and Baba's code) and I recorded the time spent by both algorithms to complete the task.

After trying the algorithms, I calculated the average time requires for all the environment configurations, so that I can analyze and compare the performance and to find out if the difference is significant. The used of different arenas configurations ensure that the statistical analysis covers the different possible performances.

3 Hypothesis

As the null hypothesis, I assume that there is no significant difference in the execution time between the two codes, suggesting that both codes have similar performance. However, the alternative hypothesis challenges this assumption and proposes that there is indeed a significant difference in the execution time. I hypothesize that there is a significant difference in the execution time between my code and my colleague's code. Specifically, I believe that my colleague's implementation performs faster than mine in terms of execution time. To test this hypothesis, I collected data by measuring the execution time for the same tasks using both implementations. Statistical methods, such as a t-test, will be employed to analyze the data and determine if there is a statistically significant difference in the execution time. By evaluating the results, I aim to gain insights into the relative performance of the two codes and understand which hypothesis will be confirmed.

4 Experiment Results

After collecting all the data regarding the execution time, all the averages were made for each configuration. I decide to make a comparison between the mean execution time of both algorithms in different configurations. Using jupyter Notbook, I got the following plots :

- Configuration 1: 4 Tokens Environment.

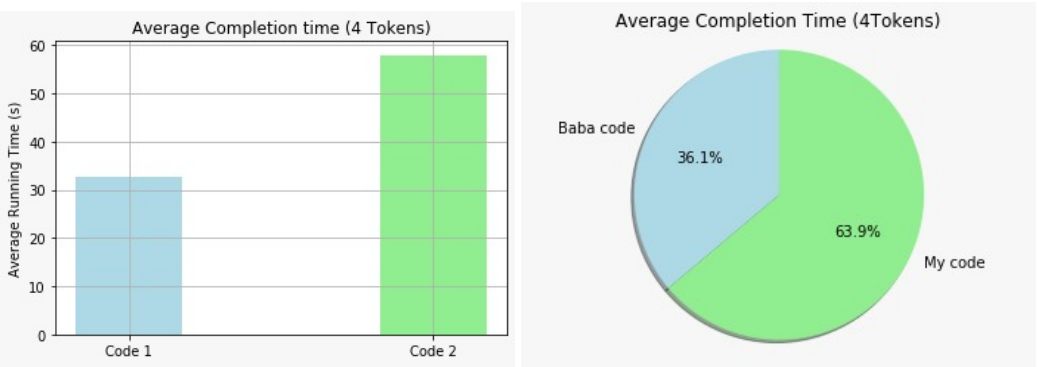

Figure 2: 4 Tokens Environment Results

- Configuration 2: 5 Tokens Environment.

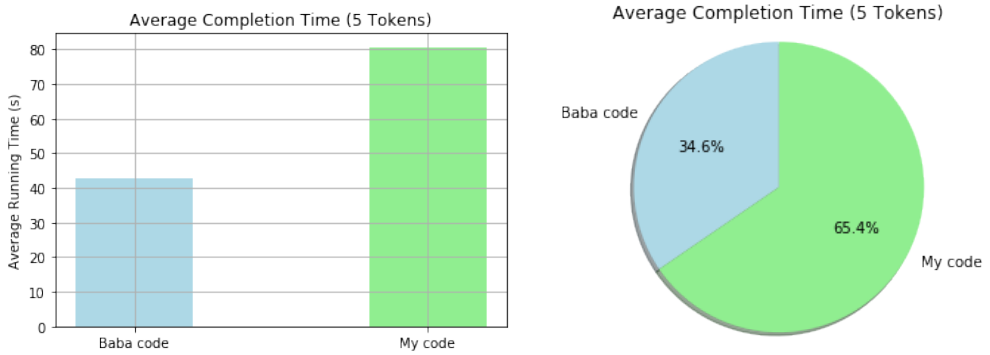

Figure 3: 5 Tokens Environment Results

- Configuration 3: 6 Tokens Environment.

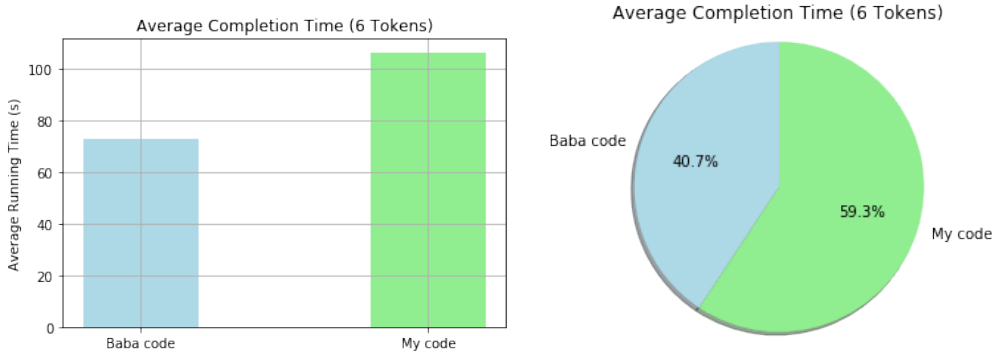

Figure 4: 6 Tokens Environment Results

- Configuration 4: 8 Tokens Environment.

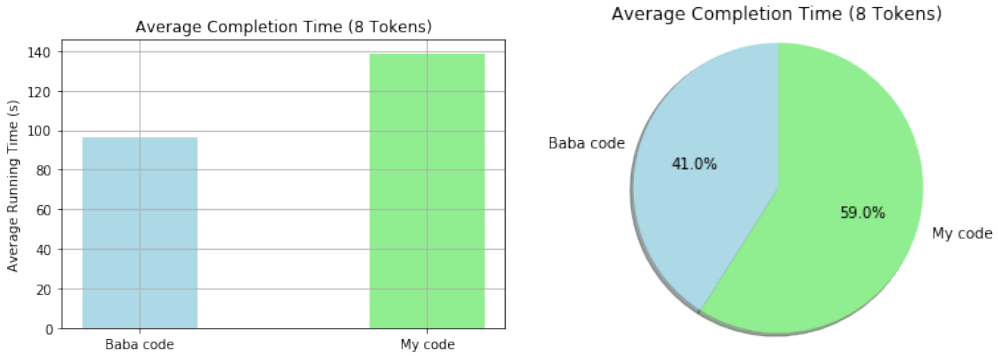

Figure 5: 8 Tokens Environment Results

This graphs show that for each configuration my robot was slower than Baba's (my colleague) robot. We notice that also, as the number of tokens increases the time difference between the two algorithms becomes smaller.

After these separate data representations, I made a one comparative plot :

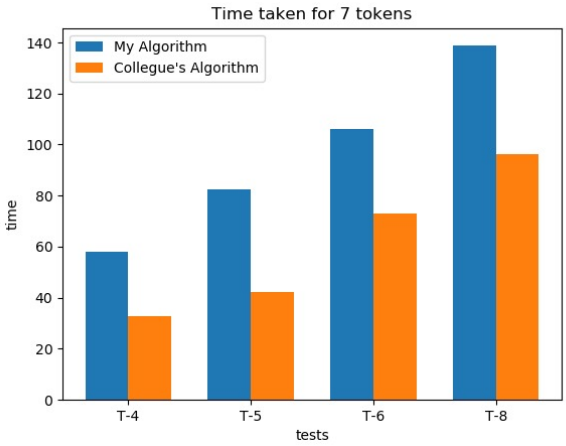

Figure 6: Test Results

In Figures 2, 3, 4 and 5, we can see two different representations of the average execution time for both algorithms in the environments presented earlier. And Figure 6 shows a comparative Bar chart collecting the results of the 4 environment.

Upon analyzing the bar charts and visualizing the overall mean results plot, it can be observed that Baba's code consistently demonstrates lower execution times compared to my code. The bar charts indicate a clear trend where Baba's code exhibits faster performance across various configurations. This suggests that Baba's code is optimized for faster execution, We can observe also regarding the relationship between the number of tokens and the difference in code implementation time. It appears that as the number of tokens increases, the disparity in execution times between Baba's algorithm and my algorithm diminishes.

It is important to conduct a statistical analysis to further validate the Observations. A suitable approach would be to perform a statistical test, such as the Lilliefors test or the t-test, to assess the significance of the observed differences in code implementation time.

5 Tests

5.1 Lilliefors Test

The Lilliefors test is a statistical test used to determine whether a given sample of data follows a normal distribution. It is similar to the Kolmogorov-Smirnov test, but unlike the latter, it does not require knowledge of the mean and standard deviation. Instead, it estimates these parameters based on the sample data and compares them to the expected values for a normal distribution. The Lilliefors test is often used in situations where it is not possible to assume that the population follows a specific normal distribution. The Lilliefors test is used to test whether a sample of data follows a normal distribution, and if the sample does follow a normal distribution, then the parametric t-test can be used. So in order to proceed for other tests we need first to perform the Lilliefors.

Code	Configuration	Test Statistic (h)	p-value
Baba's code	4 Tokens	0	0.4429
My code	4 Tokens	0	0.4229
Baba's code	5 Tokens	0	0.5033
My code	5 Tokens	0	0.4657
Baba's code	6 Tokens	0	0.0387
My code	6 Tokens	0	0.0573
Baba's code	8 Tokens	0	0.0500
My code	8 Tokens	0	0.0499

Table 1: Lilliefors Test Results

Based on the Lilliefors test results in the table, we can conclude the following: For all configurations, the test statistic (h) is 0, indicating that the data distribution does not significantly deviate from a normal distribution. So we will use the T-test.

5.2 T-Test

The T-test is a statistical method used to compare the means of one or two populations through hypothesis testing. It allows us to determine if there is a significant difference between two groups or if a single group differs from a known value. In order to compare the performance of my code and Baba's code in each environment, I employed the two-sample one-tail T-test to determine which algorithm demonstrates a significantly different execution time. The results of Lilliefors test showed that the dataset follows a normal distribution which allowed as to perform a T-test.

The test results obtained from the T-test are presented in this table:

Configuration	t-statistic	Degrees of Freedom	p-value
4Tokens	-15.442	18	1.594e-10
5Tokens	-18.395	18	4.184e-11
6Tokens	-14.467	18	4.702e-10
8Tokens	-9.824	18	2.113e-08

Table 2: T-Test Results

Based on the results of the t-tests;

- The p-values are all much smaller than the chosen significance level of 0.05, providing strong evidence to reject the null hypothesis and support the alternative hypothesis that there is a significant difference in performance times between the two codes.
- The t-statistic values obtained for all configurations were negative, indicating that the mean execution time of Baba's algorithm was consistently lower than that of my algorithm. Additionally, the p-values were extremely small (significantly less than 0.05), providing strong evidence against the null hypothesis.

6 Conclusion

In conclusion, the statistical analysis performed in this study has shed light on the comparison between my code and Baba's code in terms of performance. The results have provided evidence to support or reject the null hypothesis and validate the alternative hypothesis. Regarding the null hypothesis, the t-test results have demonstrated that there are indeed significant differences between my code and Baba's code in terms of execution time . In all configurations tested, the p-values were significantly below the chosen significance level, indicating a rejection of the null hypothesis. Therefore, the Null hypothesis (H0) was rejected.

Conversely, the alternative hypothesis, which proposes a significant difference in the execution time, has been confirmed. The t-test results have indicated that Baba's code exhibits faster execution times compared to my code in all configurations. It is important also to notice that t as the number of tokens increases the time difference between the algorithms decreases, this is due to the fact that Baba' aslgo is made based on assumption that tokens are placed in a circle, but since in my configuration I made them random, Baba's algo shows delays in 8 tokens environments and there were some failures.