UNIVERSITÀ DEGLI STUDI
DI GENOVA

RESEARCH TRACK 2

STATISTICAL ANALYSIS ASSIGNMENT 1

DIBRIS

SUBMITTED BY

Awais Tahir, 5174335

SUBMITTED TO

Prof. Carmine Recchiuto

Abstract

This report focuses on conducting a statistical analysis to compare the effectiveness of two algorithms - my own algorithm and the professor's algorithm - in completing a specific task. The task involves simulating a robot in a pygame environment, where the objective is to complete laps through a track while occasionally moving silver tokens through its lifting mechanism to a spot behind it. The analysis was conducted using Jupyter Notebook, and it pertains to the first assignment of Research Track I, based on Python Robotics Simulator.

Índice

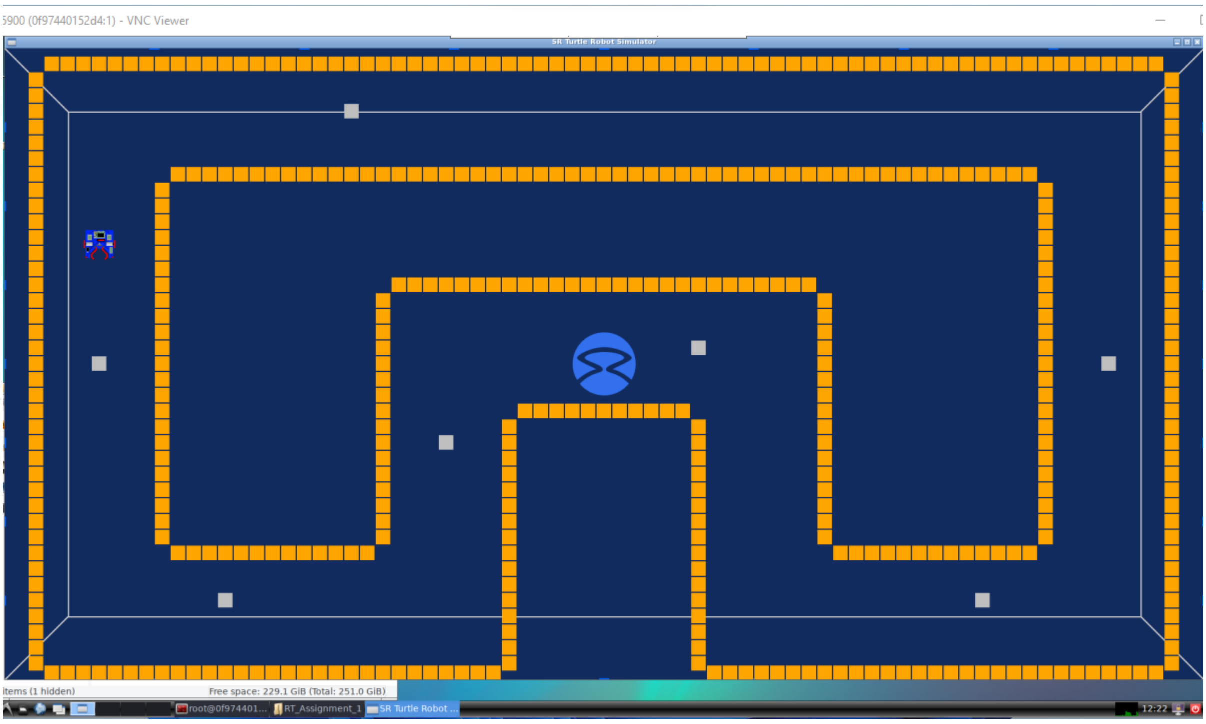
1. Introduction

1.1. Simulation Environment

Student Robotics has developed a basic and portable robot simulator, which has been adapted for use in the Research Track 1 course. The simulator features arenas that have been modified for the course's specific assignments. Within the simulator, the robot will appear in an arena made up of square tokens, with two distinct colors: silver and gold.

1)Golden tokens: Walls of the simulator are made by Golden token.

2)Silver tokens: Silver tokens are randomly placed in the simulator which robot has to collect.



2. Design and Methodology

In the python environment illustrated in figure 1, two algorithms are being compared: one provided by the professor, and one developed by a student (which will be referred to as ”Professor Code.^{a}nd ”My Code,respectively). The comparison involves directing the robot to move through the circuit in the simulated environment, while collecting and releasing silver tokens and avoiding golden tokens. The performance of the algorithms will be determined by the time taken to complete the task, assuming all other variables (such as the positions of the tokens and the robot's starting point) remain constant. To measure performance, we will record the time between each silver token grabbing and releasing event, as these events are evenly spaced within the environment and serve as an effective indicator of distance traveled, task execution efficiency, and overall circuit completion time. A significance level of 5

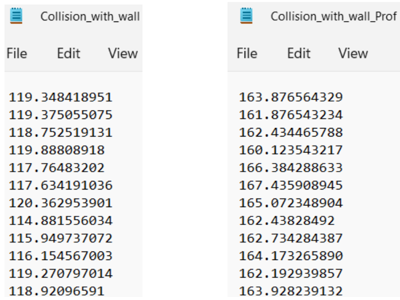
3. Implementation

To evaluate some differences between the two algorithms, the following parameters are taken into consideration: (i) Robot's change in direction (ii) Time is taken to complete the circuit (iii) Number of time simulation failed

3.1. Data Preprocessing

3.1.1. Data for the first analysis Collision with wall

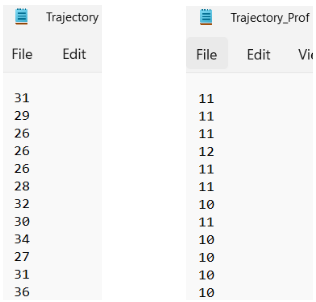
The primary goal of obtaining data for the first analysis parameter is to examine the behavior of the robot when it is in close proximity to a wall. To achieve this, modifications were made to both the Professor Code and My Code, specifically in the lines of code responsible for managing changes in direction near the walls (and therefore, close to the golden tokens). A simple modification was made by adding -1.^{ea}ch time the program reached that part of the code during runtime.



Collision_with_wall	Collision_with_wall_Prof
File Edit View	File Edit View
119.348418951	163.876564329
119.375055075	161.876543234
118.752519131	162.434465788
119.88808918	160.123543217
117.76483202	166.384288633
117.634191036	167.435908945
120.362953901	165.072348904
114.881556034	162.43828492
115.949737072	162.734284387
116.154567003	164.173265890
119.270797014	162.192939857
118.92096591	163.928239132

3.1.2. Data for the second analysis Trajectory

The primary objective in obtaining data for the second analysis parameter is to track the time as soon as the program enters the section of code responsible for robot movement (typically located within the main() function). The code is designed to record the end time once the robot successfully grabs the 8th silver token (due to the addition of one token). Although the circuit is not a true representation of a real-world scenario, the fact that the final token's position is fixed makes time an effective parameter for analysis.



Trajectory	Trajectory_Prof
File Edit	File Edit Vi
31	11
29	11
26	11
26	12
26	11
28	11
32	10
30	11
34	10
27	10
31	10
36	10

3.1.3. Additional Tokens

When considering the two algorithms (My Code and Professor Code), we navigated to the directory .../sr/robot and located a file called sunny$_{s}ide_{u}p_{a}rena.py.Thisfilecontainscodeforgeneratingthemap, includingth$

3.1.4. Hypothesis

There are three hypotheses to be considered:

Hypothesis 1:

Hypothesis 1 arises from an experimenter's poor choice of participants, where they select a sample that does not accurately represent the population under analysis.

This is known as a biased sample. For example, if we want to draw conclusions about the entire population of Italian students and we select a sample consisting only of Robotics students who are predisposed to finding bugs in algorithms, our sample would be biased.

Hypothesis 2:

Hypothesis 2 is attributed to random variability. If we were to repeat the experiment with a different sample from the same population, we may not obtain the same results due to various factors, both known and unknown, that are influenced by chance.

To account for the effects of chance, statistical tests can be employed to determine the significance of the results.

Hypothesis 3:

The hypothesis can only be accepted after excluding hypotheses 1 and 2.

Two hypotheses, Ha (alternative hypothesis) and Ho (null hypothesis), can be used to compare the performance of the two algorithms. To distinguish between the two hypotheses, it is important to understand their nature:

Ho: The robot from both My Code and Professor Code complete the circuit in a similar amount of time, indicating that their means are roughly equal.

Ha: The robot from both My Code and Professor Code are not able to complete the circuit in a similar amount of time, indicating that their means are not equal.

It should be noted that the acceptance of any hypothesis is dependent on the exclusion of the first and second hypotheses.

3.1.5. Observation Data

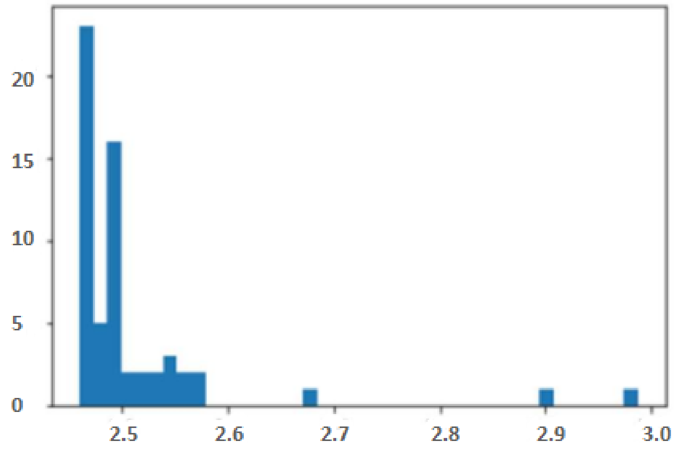
A total of 60 observations were recorded for each code, measuring the time (in seconds) taken for the robot to complete its tasks. The data was obtained automatically by including instructions in the Python script to save time data. Presented below are the first 5 observations for My Code and Professor Code, respectively.

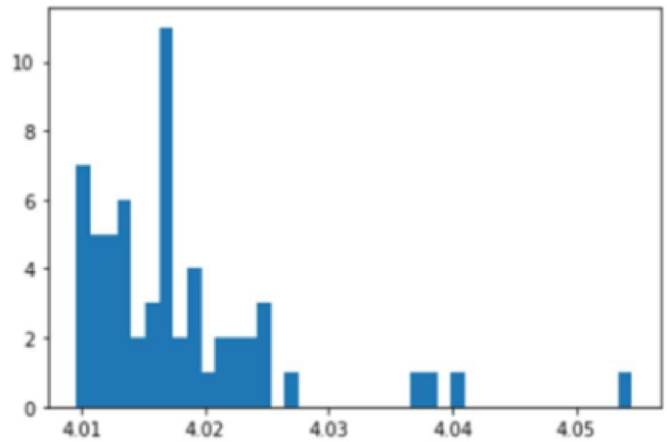
```
collision_with_wall_mean = np.mean(collision_with_wall)
timestamp_mean = np.mean(timestamp)
collision_with_wall_Prof_mean = np.mean(collision_with_wall_Prof)
timestamp_Prof_mean = np.mean(timestamp_Prof)

# Calculating the standard deviation
collision_with_wall_std = np.std(collision_with_wall)
timestamp_std = np.std(timestamp)
collision_with_wall_Prof_std = np.std(collision_with_wall_Prof)
timestamp_Prof_std = np.std(timestamp_Prof)
```

	time(s)
0	2.556254
1	2.514890
2	2.658965
3	2.591450
4	2.538406

	time(s)
0	4.009535
1	4.019174
2	4.054525
3	4.010038
4	4.014200

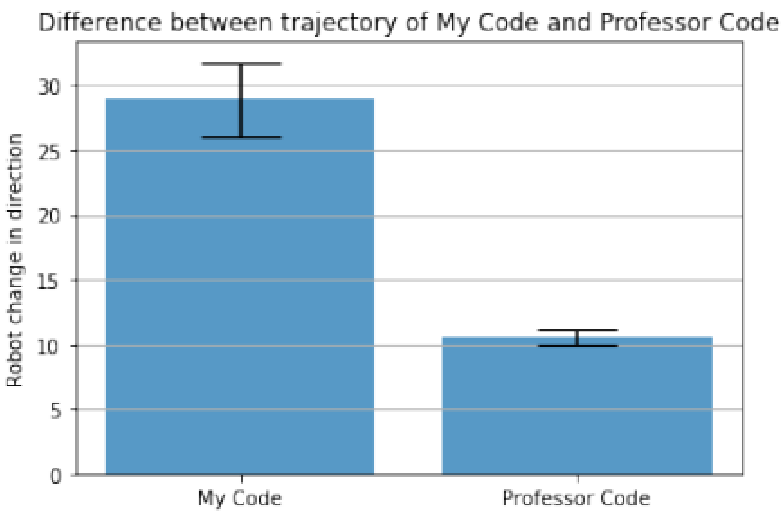
It is immediately apparent that my code has time superiority over professor code, exhibiting lower run times and hence completing the circuit faster.

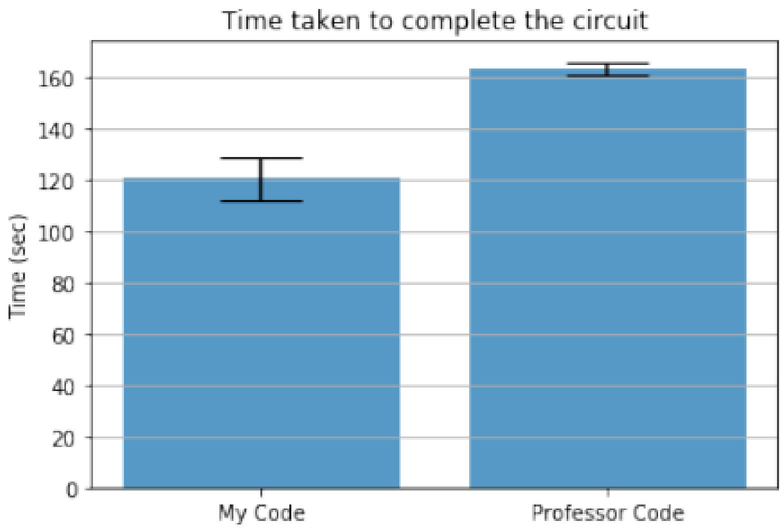


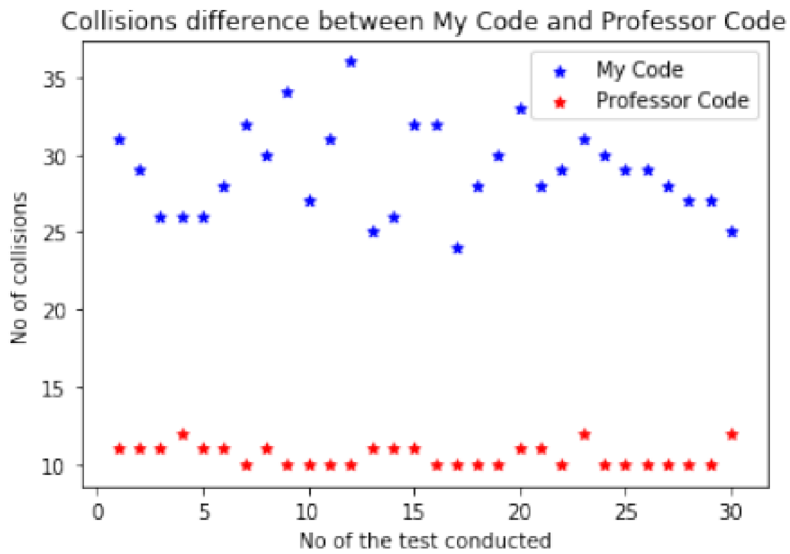


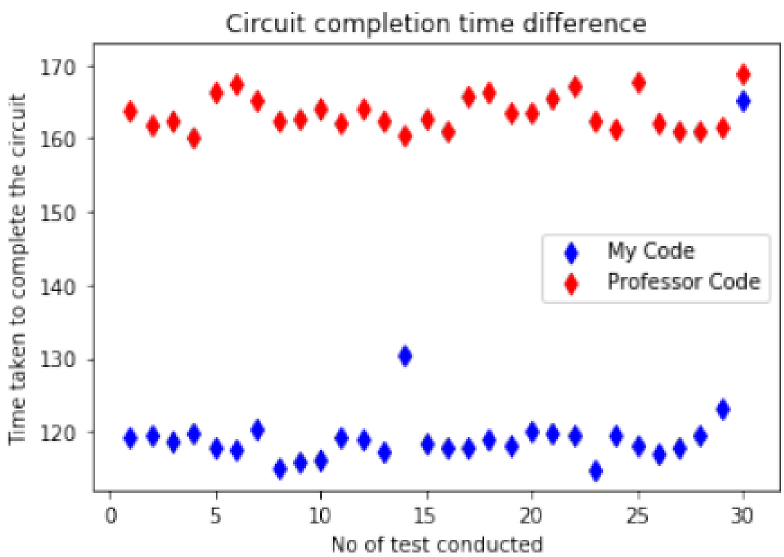
4. Results

This section presents the experimental results that analyze the performance of two algorithms, namely My Code and Professor Code. The figure below illustrates the difference in trajectory between these two codes.




Time taken to complete the circuit
Time (sec)
160
140
120
100
80
60
40
20
0
My Code
Professor Code


Collisions difference between My Code and Professor Code
My Code
Professor Code
No of collisions
35
30
25
20
15
10
0
5
10
15
20
25
30
No of the test conducted


Circuit completion time difference
Time taken to complete the circuit
170
160
150
140
130
120
My Code
Professor Code
0
5
10
15
20
25
30
No of test conducted

The figure below shows the Graphing Representation of the collision difference of both the two codes (my code professor code).

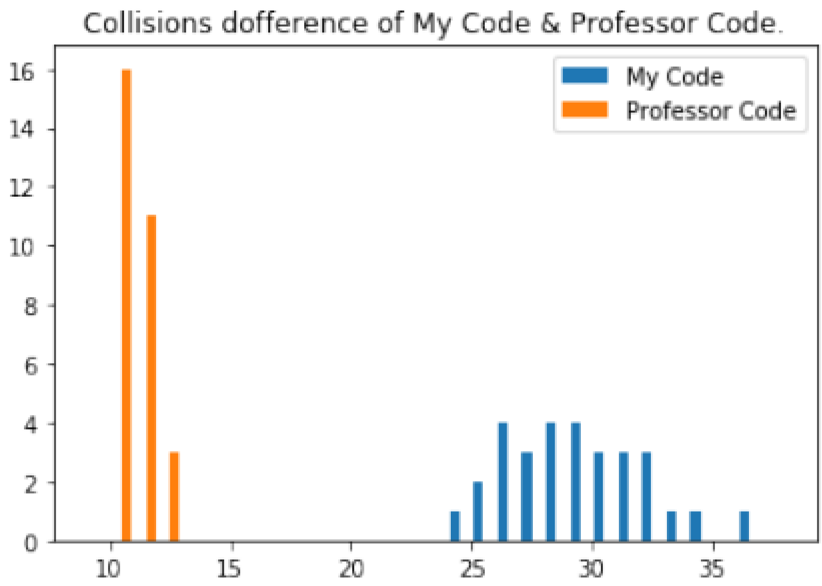
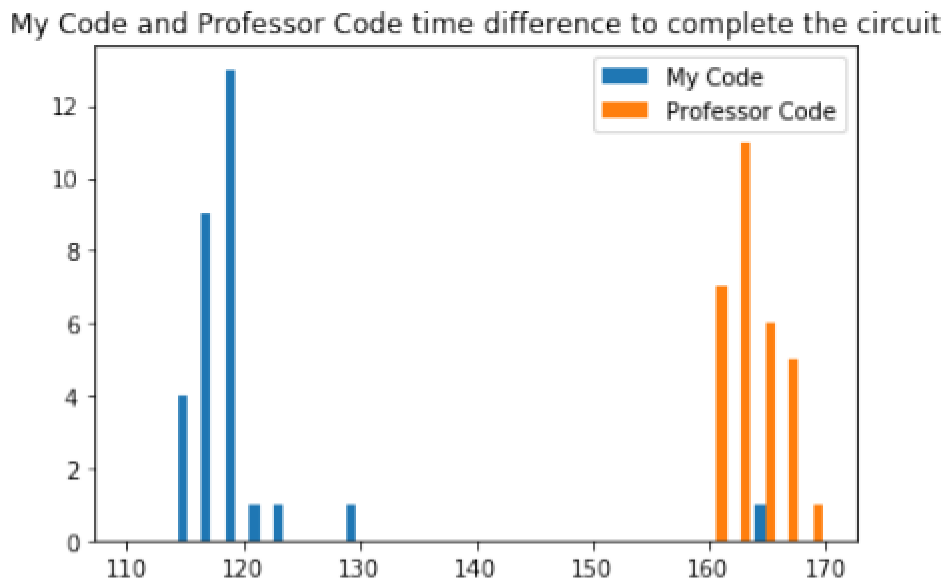


Figure below shows the Graphing Representation of Time taken to complete the circuit by my code professor code.



4.1. Analysis

4.1.1. Chi-Square Test

We compare two percentages obtained in a controlled experiment to determine if the observed difference is statistically significant or due to chance.

We begin by stating the null hypothesis, or H0:

• H0: There is no significant difference between the percentages in the population (the observed difference is due to chance).

• Next, we perform a statistical test to either accept or reject H0.

• The formulation of the hypothesis can depend on various factors and may be more complex. However, we always assume this type of hypothesis in the following analysis.

The decision to accept or reject the null hypothesis is based on the nature of the dependent variable. When comparing two percentages, a Chi-square test is appropriate.

4.1.2. T-Test Analysis

The T-Test, also known as Student's Test, utilizes the t-distribution and is a suitable test for determining the significance of a sample mean or the difference between the means of two samples, particularly in the case of small samples when the population variance is unknown (in which case the sample variance is used as an estimate of the population variance). The relevant test statistic, t, is computed from the sample data and then compared to its expected value based on the t-distribution at a specific level of significance and degrees of freedom to accept or reject the null hypothesis (source: https://2021.aulaweb.unige.it/pluginfile.php/422414/mod_resource/content/0/04_Statistics.pdf).

In my Jupyter Code, I performed two T-Tests between the two datasets and expected them to be significantly different. This test is used to test the null hypothesis that two independent samples have the same mean values. By default, this test assumes that the data have identical variances to determine the degree of difference between them. The results of the test are as follows:

Analysis I:
Collision_Static = 34.031085935665956
p_Values = 5.05537028942152e-40
Analysis II:
Collision_Static = -25.589230325578303
p_Values = 2.750043113035198e-33

Where the p_value is less than our significance value and with a negative t-value, the null the hypothesis will be rejected and the alternative sustained.

5. Conclusion

The main objective of the statistical analysis in this project is to identify significant differences between my code and the professor's code that affect the robot's behavior in the environment. To ensure consistency in the data, I took multiple observations under the same conditions, considering that the results are non-deterministic. The analysis showed that there are substantial differences in lap times and distances from golden tokens between my robot and the professor's robot. My robot maintains a higher distance from the walls, which can prevent wall-crush incidents. Moreover, despite the number of silver tokens, my robot completes laps faster than the professor's robot. These initial assumptions were confirmed by the results of the statistical tests conducted.