Research Track 2 - Statistical Analysis of RT1 Assignment 1

Introduction

In this report it will be presented the statistical analysis of the first assignment of Research Track I course, which aims to compare the performances obtained by the script I personally developed and the professor's one.

For the experiment, I have considered two different arenas (i.e. the same arena but with different tokens displacement):

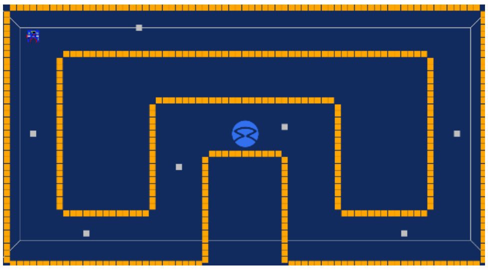

Figure 1: Arena 1

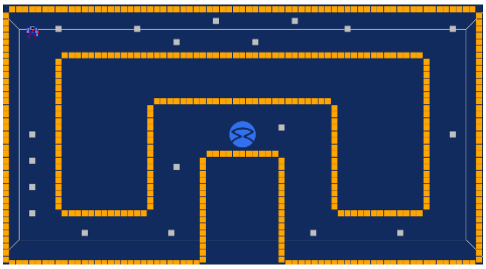

Figure 2: Arena 2

For each arena I ran both scripts five times. The number of laps (and the number of tokens to be grabbed) depends on the type of the arena:

- **Arena 1**: 7 tokens, 5 laps per simulation.
- **Arena 2**: 19 tokens, 3 laps per simulation.

Please note that a simulation is considered valid even if at some point the robot crashes or makes something wrong (e.g. changing direction), in that case, only the unaffected data will be collected.

The analysis is made up of three parts: **Data Collection**, **Data Processing** and **Data Interpretation**, and will be described in the sections below.

Data Collection

The first part of a statistical analysis consists of collecting raw data from the source code of our scripts. To do that, I implemented a simple python Class, **Statistics.py**, which can retrieve data and append them on *.txt* files. Specifically, anytime the user runs the simulation, two text files will be created:

- *distances[i][j].txt*: each column contains the distances from the walls respectively for the left, frontal and right portion of the robot view. In the name of the file, "i" indexes the script (script 1,2), "j" indexes the occurrence (i.e. the current simulation, from 1 to 5).
- *lapTime[i][j].txt*: there is only one column which contains the time interval between two grabbed tokens. The first element of the "array" is the number of tokens in the circuit. The indexes in the name of the file follow the same logic as the first *.txt* file.

The result consists of two folders, one for each arena, with a couple of *.txt* files for each occurrence.

Finally, the second part of the data collection consisted of annotating the performance of each occurrence, i.e. if the robot has completed the expected laps or not. The table containing this kind of info will be shown in the next paragraphs.

Data Processing

The processing of data has been done on **Matlab**. More precisely, I implemented a Matlab script, **RT2_statistic.m**, on which I imported the collected data. Data are now preprocessed by replacing all values equals to 100 (which means that the laser was out of range) with *NaN* values.

After this preliminary step, I started processing data to obtain some interesting plots that can point out the difference in the performances between the two scripts. Either for the values of distances and temporal aspects, I considered as object of analysis the performance obtained in each occurrence, which can be considered a random instance of the code.

Concerning the distance values, for each group I considered the mean and the standard deviation of the *right*, *frontal* and *left* distances. I also retrieved the percentage of time in which the robot is less than 0.8m away from the left and right walls. This can be easily computed by counting the number of values less than the threshold and dividing it by the total number of distance values in that occurrence.

For the data related to the second *.txt* file (*LapTime.txt*), I reshaped each array (which contains the timestamp at each token grabbed) into multiple rows, so that each one represents an entire lap. In this way I can compute the time taken to complete each lap and consequently the mean and standard deviation for each occurrence of the script.

Data Interpretation

For this section, I created a table and some plots to visualize the processed data (more precisely, there are five plots for each type of arena).

Here's the table containing the performance of each occurrence:

Occurrence	Normal Arena (7 tokens a lap - 5 laps)		Modified Arena (19 tokens a lap - 3 laps)	
	Script 1	**Script 2**	**Script 1**	**Script 2**
1	OK	C(22)	OK	X(38)*
2	OK	OK	C(22)	X(37)
3	OK	X(21)	OK	X(38)
4	OK	X(23)*	OK	OK
5	OK	OK	OK	C(20)

Table 1: table that shows the performances of the two scripts in Normal Arena and in Modified Arena. ('OK' means that the robot has completed all the laps, 'C(i)' means that the robot changed direction after grabbing the i-th token, 'X(i)' means that the robot became stuck after grabbing the i-th token) ('' means that the robot hit the walls without affecting the rest of the test).*

As we can see, in the first script the robot doesn't seem to show any relevant issue in completing the expected laps, while in the second script it tends to become stuck or to change direction more frequently, especially when it must grab the last token of the circuit.

The first plot is a barplot which represents the average golden token distance for the three considered orientations:

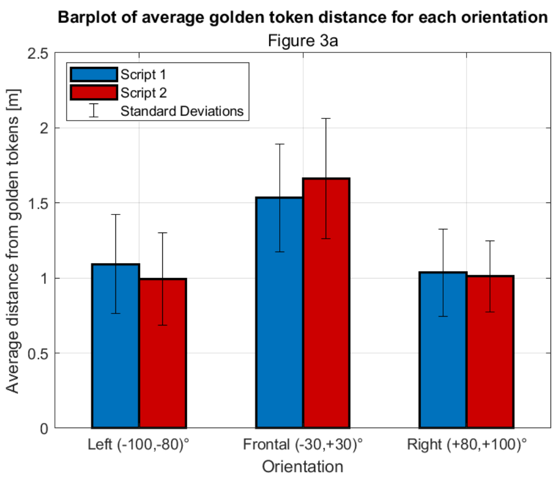

Figure 3a: Barplot of average golden token distance for each orientation (ARENA 1)

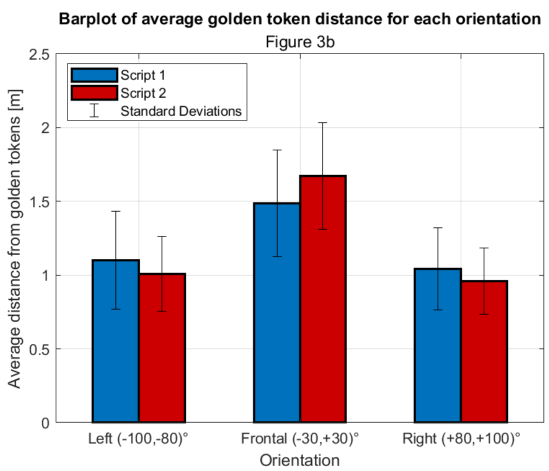

Figure 3B: Barplot of average golden token distance for each orientation (ARENA 2)

This plot shows that the average distance between the robot and the closest golden token in the three directions is similar and it is interestingly similar also between the two arenas.

To have a more intuitive idea of how the robot is located with respect to the walls, I created a set of violin plots, respectively for the left and the right field of view of the robot:

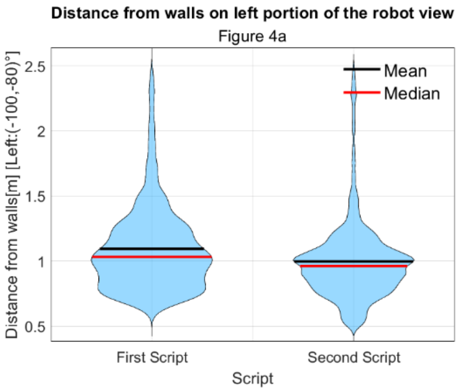

Figure 4a: Violin plot of the distribution of the distances (LEFT) (ARENA 1)

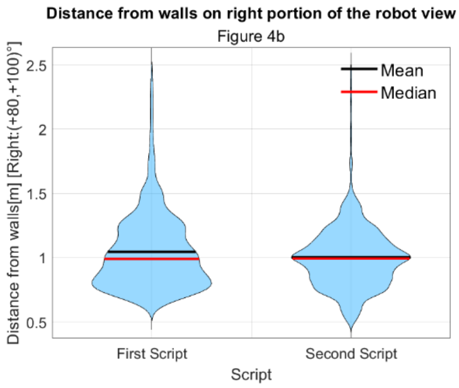

Figure 4b: Violin plot of the distribution of the distances (RIGHT) (ARENA 1)

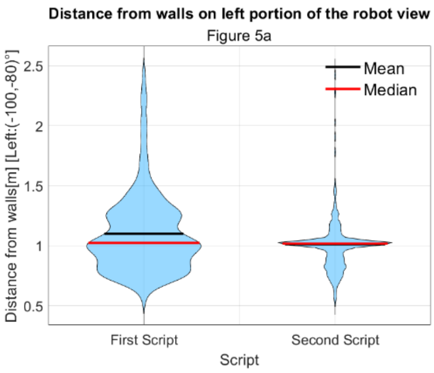

Figure 5a: Violin plot of the distribution of the distances (LEFT) (ARENA 2)

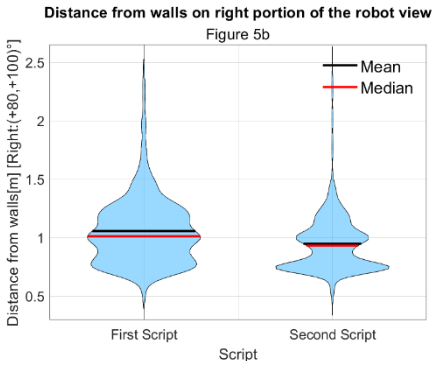

Figure 5b: Violin plot of the distribution of the distances (RIGHT) (ARENA 2)

The results for the Arena 1 show that the shape generated by the two scripts is slightly different (for both directions), the first script has a bigger standard deviation, which it is denoted by a more elongated shape (*Fig.4a,4b*). This also holds for the Arena 2 (*Fig.5a,5b*)., but in this case the differences are more explicit, especially in Left direction (*Fig. 5a*) where the shape generated by the second script is very squeezed towards its mean value.

Here are the results for **Shapiro-Wilk** Test, which one of the most powerful tests of Normality for small samples and let us know if the 5 samples related to the distance from walls belong to a *Normal Distribution*:

- **Arena 1**:
  - Script 1: mean distance between the robot and the ***Left*** walls: H=0, p=0.2787
  - Script 1: mean distance between the robot and the ***Right*** walls: H=0, p=0.0952
  - Script 2: mean distance between the robot and the ***Left*** walls: H=0, p=0.3441
  - Script 2: mean distance between the robot and the ***Right*** walls: H=0, p=0.7272
  - **Results**: the null hypothesis is that the population is normally distributed and it's not possible to reject it in each of the hereabove tests with a level of significance 0.05, therefore it is possible to proceed with the T-Tests. *However*, the p-value of these tests should cast doubt on the validity of the null hypothesis.
- **Arena 2**:
  - Script 1: mean distance between the robot and the ***Left*** walls: H=0, p=0.8023
  - Script 1: mean distance between the robot and the ***Right*** walls: H=0, p=0.9182
  - Script 2: mean distance between the robot and the ***Left*** walls: H=0, p=0.5833
  - Script 2: mean distance between the robot and the ***Right*** walls: H=0, p=0.6048

- **Results**: the null hypothesis is that the population is normally distributed and it's not possible to reject it in each of the hereabove tests with a level of significance 0.05, therefore it is possible to proceed with the T-Tests.

Here's a **two samples T-Test** for each of the arenas about the difference between the two groups of the mean distance value from walls for the lateral portions of the robot view (i.e. *Left* and *Right*).

For each test, The *Null Hypothesis* $\mathbf{H_0}$ and the *Alternative Hypothesis* $\mathbf{H_1}$ are:

- **H0**: $\mu 1 = \mu 2$ (the two population means are equal)
- $\mathbf{H_1}$ (right-tailed): $\mu_1 > \mu_2$ (population 1 mean is greater than population 2 mean)

Here are the results:

- **Arena 1**:
  - The value of the mean distance between the robot and the ***Left*** walls for the 5 occurrences performed by the first script (M = 1.0899 [m], SD = 0.0136 [m]) is higher than the same value but related to the 5 occurrences performed by the second script (M = 0.9932 [m], SD = 0.0181 [m]), $t(8) = 2.306$, $p < 0.01$.
  - There are no significative differences between the value of the mean distance between the robot and the ***Right*** walls for the 5 occurrences performed by the first script (M = 1.0353 [m], SD = 0.0155 [m]) and the same value but related to the 5 occurrences performed by the second script (M = 1.0121 [m], SD = 0.0256 [m]), $t(8)$=2.306, p =.0608. If we increase the significance level up to 10%, the test shows that the value related to the first script is higher than the one in the second script.

- **Arena 2**:
  - The value of the mean distance between the robot and the ***Left*** walls for the 5 occurrences performed by the first script (M = 1.1008 [m], SD = 0.0074 [m]) is higher than the same value but related to the 5 occurrences performed by the second script (M = 1.0088 [m], SD = 0.0146 [m]), $t(8) = 2.306$, $p < 0.01$.
  - The value of the mean distance between the robot and the ***Right*** walls for the 5 occurrences performed by the first script (M = 1.0418 [m], SD = 0.0131 [m]) is higher the same value but related to the 5 occurrences performed by the second script (M = 0.9603 [m], SD = 0.0691 [m]), $t(8)$=2.306, p =.0160

The last plot related to the distance between the robot and the golden tokens expresses the percentage of time in which the robot is close to a wall, with a threshold of *0.8 [m]:*

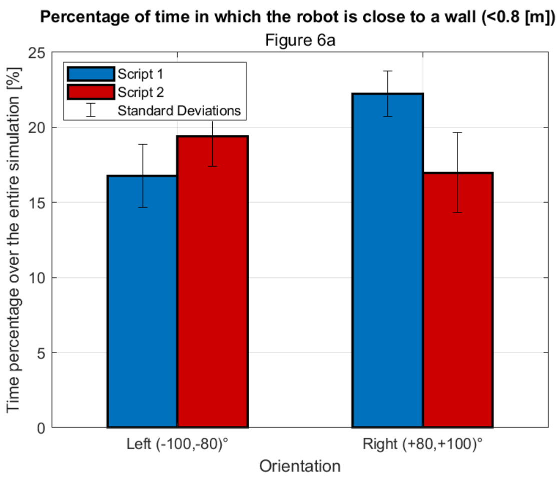

Figure 6a: Barplot of percentage of time in which the robot is close to a wall (ARENA 1)

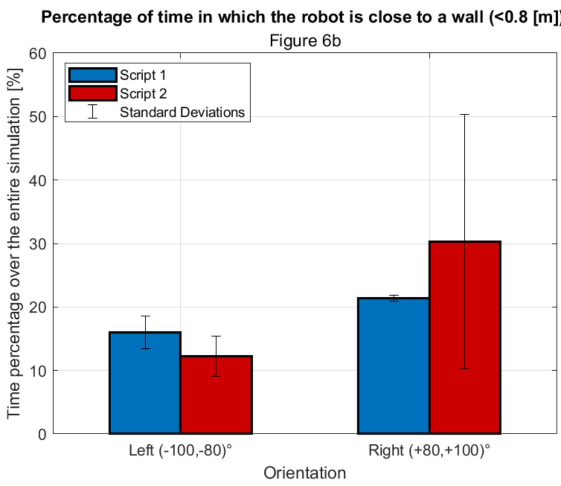

Figure 6b: Barplot of percentage of time in which the robot is close to a wall (ARENA 2)

The barplot for *Arena 1* (*Fig. 6a*) shows that the percentage of time in which the robot is close to a wall in the left portion of the robot view is on average lower for the first script, while for the right portion is higher.

For *Arena 2* (*Fig. 6b*), the results suggest the opposite trend for each portion of the robot view: a higher time percentage in the left portion of view and a lower one in the right portion. Please note that the large standard deviation of the latter is due to the presence of two outliers which have a percentage value of about 50%.

Finally, the last plot is about the average time to complete the circuit:

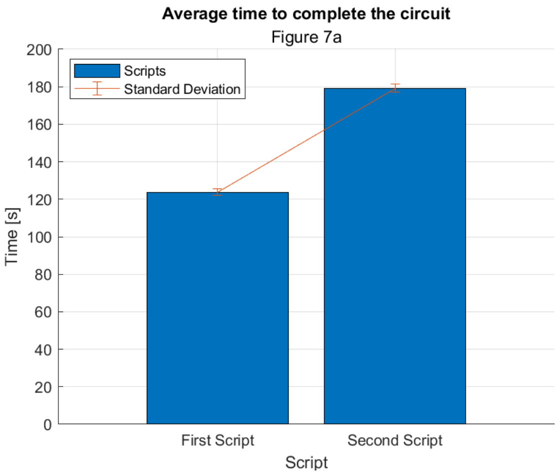

Figure 7a: Barplot of average time to complete the circuit (ARENA 1)

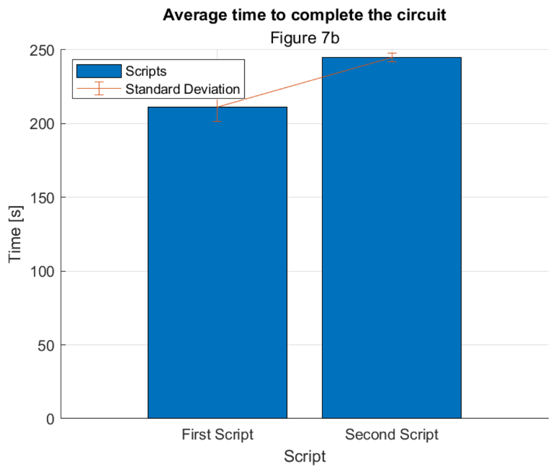

Figure 7b: Barplot of average time to complete the circuit (ARENA 2)

In the first case (*Fig. 7a*), the result may suggest that the robot of the first script is faster than the one in the other script, the former requires on average 124 [s] to complete a lap while the latter requires 177 [s]. In the second case, the first robot still requires less time, but their difference is decreased: 213 [s] for script 1 and 243 [s] for script 2. This could be due to that, when the robot fails to grab the token, the first one wastes several seconds because it tries to re-orientate itself until the token is grabbed, while the second one would cause the end of the simulation.

Here's the results for **Shapiro-Wilk** Test, which checks if the samples related to the average time to complete the circuit belong to a *Normal Distribution*:

- **Arena 1**:
  - Script 1: average time to complete the circuit (5 samples): H=0, p=0.4055
  - Script 2: average time to complete the circuit (5 samples): H=1, p=0.4252

  - **Results**: the null hypothesis is that the population is normally distributed and it's not possible to reject it in both tests with a level of significance 0.05, therefore it is possible to proceed with the T-Tests. *However*, the p-value of these tests should cast doubt on the validity of the null hypothesis.

- **Arena 2**:
  - Script 1: average time to complete the circuit (5 samples): H=0, p=0.1276
  - Script 2: average time to complete the circuit (5 samples): H=1, p=0.0338

  - **Results**: the null hypothesis is that the population is normally distributed and it's possible to reject it for Script 2 at level of significance 0.05. For Script 1 the null hypothesis can't be rejected but the small value of p cast doubt on its validity, therefore it is not possible to proceed to T-Tests.

Here's a **two samples T-Test** for Arena 1 related to the difference between the two scripts about the average time to complete the circuit:

The *Null Hypothesis* $\mathbf{H_0}$ and the *Alternative Hypothesis* $\mathbf{H_1}$ are:

- **H0**: $\mu1 = \mu2$ (the two population means are equal)
- $\mathbf{H_1}$ (right-tailed): $\mu_1 > \mu_2$ (population 1 mean is greater than population 2 mean)

- **Arena 1**:
  - The average time to complete the circuit for the robot of the first script (M =123.82[s], SD = 1.89 [s]) is <u>lower</u> than the one required by robot of the second script (M = 179.25 [s] SD = 2.08 [s]), $t(8) = 2.306$, $p < 0.01$.