UNIVERSITÀ DEGLI STUDI DI GENOVA

DIBRIS

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

RESEARCH TRACK 2

Third Assignment

Statistical Analysis

Authors:	*Student ID:*
Ami Sofia Shimizu Quijano	s6193847
Tomoha Neki	s6344955

Professor:

Carmine Recchiuto

May 30, 2024

Contents

1 Assignment Description **2**

2 Hypotheses **2**

3 Description and Motivation of Experimental Setup **2**

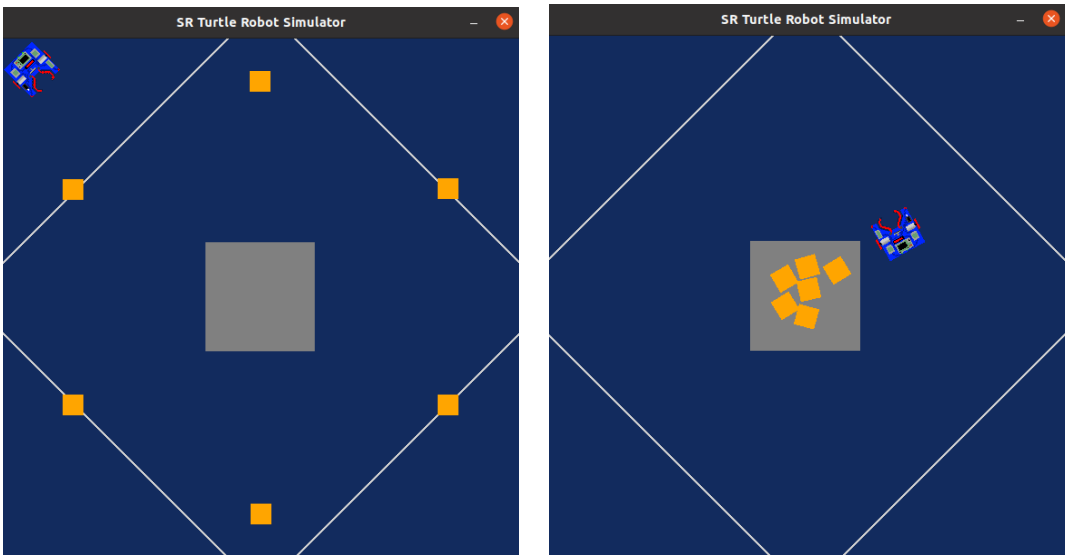
4 Results **4**

5 Discussion of Results with Statistical Analysis **5**

6 Conclusion **7**

1 Assignment Description

The following assignment consists on carrying out a statistical analysis to compare the performance of two different algorithms designed to drive a mobile robot which picks up tokens placed in a simulation environment and gathers them, as shown in Figure 1.



(a) (b)

Figure 1: a) Initial state of simulation: Tokens placed randomly in environment. b) Final state of simulation: Tokens gathered together.

The algorithm of Tomoha Neki will be referred to as *Algorithm 1*, whereas the algorithm of Ami Quijano will be called *Algorithm 2*.

2 Hypotheses

Two performance metrics are wished to be tested:

- Run time: Time taken for the robot to collect all tokens together.
- Number of failures: Number of times the robot is unable to collect all tokens together.

For each of the performance metrics, the Null hypotheses (H_0) and the Alternative hypotheses (H_a) are:

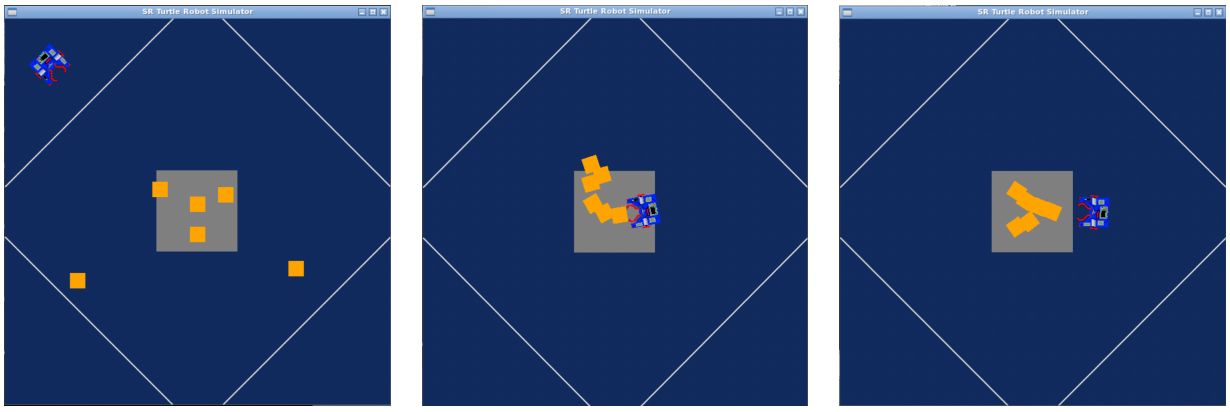
1. **Run Time**
   - $H_0 : \mu_1 = \mu_2$ (Algorithm 1 performs equally well as Algorithm 2)
   - $H_a : \mu_1 < \mu_2$ (Algorithm 1 is faster than Algorithm 2)
2. **Number of Failures**
   - $H_0 : p_1 = p_2$ (Algorithm 1 performs equally well as Algorithm 2)
   - $H_a : p_1 < p_2$ (Algorithm 1 fails less than Algorithm 2)

3 Description and Motivation of Experimental Setup

Considering the performance metrics wished to test (run time and number of failures), the experiment was designed and developed as follows:

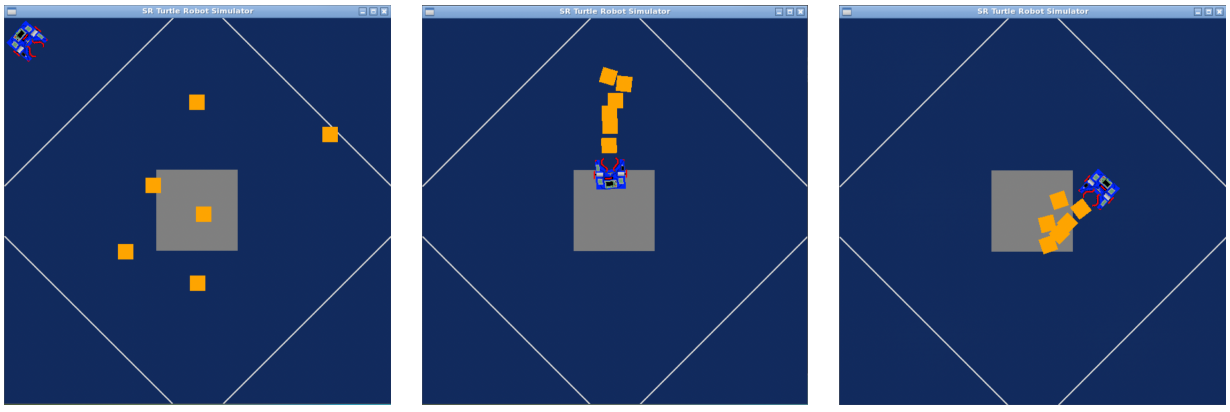
- The different random token placements in the simulation environment were obtained by scaling the radius of each token by a random number between 0 and 1 while keeping the angle as in the original version. This resulted in the tokens being equally spaced angularly but randomly displaced radially. By doing so, it was possible to have varying token positions while minimizing conflicts between the blocks. These modifications were done in the script `two_colors_assignment_arena.py`.

- The number of tokens considered in all the runs were 6.
- A test environment was developed in order to run and time multiple algorithms through a desired number of runs. For the purpose of the statistical analysis, we decided to do 50 runs of each algorithm. This decision was taken since, according to the Central Limit Theorem, for sufficiently large sample sizes (typically n > 30), the sampling distribution of the sample mean tends to be normally distributed regardless of the population distribution. Additionally, we hoped that with this number of runs there would be enough failures (> 10) to perform a Chi-Square Test as well as enough successes (> 30) that could be timed.
- In each set of runs, both algorithms ran on the same environment (placement of tokens). This condition allows to perform a Paired T-Test which is statistically more powerful in situations where pairing is possible and reduces variability compared to the Two Sample T-Test, leading to more precise estimates.
- Each set of runs had a different and random placement of tokens.
- Runs were timed using the Python function `time.time()`.
- If a run takes more than 500 seconds, it is stopped and that run is considered a failure. This decision was made after testing and ensuring that for both algorithms, a run time greater than 500 seconds represented a situation in which the robot is indefinitely stuck, which occurs when there is a token between the robot and the token wished to grab.
- All runs were done on the same laptop running Ubuntu 22.04 with no other processes running in the background.



(a) (b) (c)

Figure 2: Example of experiment. a) Initial state of simulation b) Final state of Algorithm 1 c) Final state of Algorithm 2



(a) (b) (c)

Figure 3: Example of experiment. a) Initial state of simulation b) Final state of Algorithm 1 c) Final state of Algorithm 2

4 Results

After performing 50 pairs of runs for both algorithms, each pair with a random environment, the following results were obtained. Table 1 shows the runtime for each algorithm. If the runtime exceeded 500 seconds, it was considered a failure and recorded as "-1". Table 2 shows whether each algorithm succeeded or failed in each run, with "SUCCESS" indicating the task was completed within 500 seconds and "FAILURE" indicating a timeout.

Run number	Algorithm 1	Algorithm 2
1	138.0543456	318.9028852
2	152.1015747	304.4452786
3	99.52037024	334.5002985
4	151.1789021	312.3957634
5	114.6054552	274.3943777
6	215.8103986	313.4142752
7	112.1142957	402.1961331
8	148.6598487	290.3641605
9	146.1624455	-1
10	149.6240342	291.8818815
11	146.6564169	329.4828367
12	297.2014318	274.9674318
13	152.1488256	270.3810124
14	115.1343913	270.3849983
15	155.5894198	313.3392942
16	405.4833677	339.8832877
17	184.6442544	322.8176975
18	173.6152842	321.3325841
19	116.0371039	269.8869979
20	133.0261295	-1
21	403.9426193	350.8998804
22	145.5398858	315.8995111
23	132.1090143	316.3654263
24	147.5533206	310.8322659
25	140.0441511	289.3754609
26	148.5859799	273.8209417
27	148.1272612	400.3326094
28	159.1656756	320.9897439
29	145.5891399	271.8324771
30	-1	278.9141457
31	158.6353071	270.3475635
32	155.6163764	335.4172583
33	126.6005828	311.4435296
34	146.62609	327.1177959
35	174.1569817	325.9297333
36	139.6362522	380.4817438
37	157.6228476	348.4538329
38	169.646229	294.3979769
39	135.5909698	335.9437046
40	112.0925634	291.9122658
41	180.6716633	309.0799878
42	149.6013205	274.3581386
43	153.1447635	267.8616531
44	139.6289563	286.4070084
45	204.6915617	272.3684094
46	142.5906134	270.8283823
47	167.6353343	353.4373901
48	152.6202402	284.5009818
49	156.6197202	-1
50	126.5780046	287.8937998

Table 1: Runtime of each algorithm [s]

Run number	Algorithm 1	Algorithm 2
1	SUCCESS	SUCCESS
2	SUCCESS	SUCCESS
3	SUCCESS	SUCCESS
4	SUCCESS	SUCCESS
5	SUCCESS	SUCCESS
6	SUCCESS	SUCCESS
7	SUCCESS	SUCCESS
8	SUCCESS	SUCCESS
9	SUCCESS	FAILURE
10	SUCCESS	SUCCESS
11	SUCCESS	SUCCESS
12	SUCCESS	SUCCESS
13	SUCCESS	SUCCESS
14	SUCCESS	SUCCESS
15	SUCCESS	SUCCESS
16	SUCCESS	SUCCESS
17	SUCCESS	SUCCESS
18	SUCCESS	SUCCESS
19	SUCCESS	SUCCESS
20	SUCCESS	FAILURE
21	SUCCESS	SUCCESS
22	SUCCESS	SUCCESS
23	SUCCESS	SUCCESS
24	SUCCESS	SUCCESS
25	SUCCESS	SUCCESS
26	SUCCESS	SUCCESS
27	SUCCESS	SUCCESS
28	SUCCESS	SUCCESS
29	SUCCESS	SUCCESS
30	FAILURE	SUCCESS
31	SUCCESS	SUCCESS
32	SUCCESS	SUCCESS
33	SUCCESS	SUCCESS
34	SUCCESS	SUCCESS
35	SUCCESS	SUCCESS
36	SUCCESS	SUCCESS
37	SUCCESS	SUCCESS
38	SUCCESS	SUCCESS
39	SUCCESS	SUCCESS
40	SUCCESS	SUCCESS
41	SUCCESS	SUCCESS
42	SUCCESS	SUCCESS
43	SUCCESS	SUCCESS
44	SUCCESS	SUCCESS
45	SUCCESS	SUCCESS
46	SUCCESS	SUCCESS
47	SUCCESS	SUCCESS
48	SUCCESS	SUCCESS
49	SUCCESS	FAILURE
50	SUCCESS	SUCCESS

Table 2: Success/Failure of each algorithm

5 Discussion of Results with Statistical Analysis

The following statistical analysis was performed to evaluate the performance metrics (runtime and number of failures) according to the initially made hypotheses.

1. **Run Time**

According to the obtained results, a statistical test for evaluating the runtime can be done by considering the 46 paired measurements which remain after discarding the runs where either of the algorithms failed. The appropriate statistical test to evaluate the runtime in this case is the Paired T-Test since

- With a sample size of 46 (> 30) for both algorithms, it can be assumed that the sample distribution of both algorithms is normal.
- The observations of *Algorithm 1* can be paired with those of *Algorithm 2* since in each set of runs both algorithms were tested with the same token placement in the environment.

The Paired T-Test is done as follows:

a. Determine the hypotheses in terms of the difference, d, between the paired measurements.

   - $H_0 : \mu_d = 0$ (Algorithm 1 performs equally well as Algorithm 2)
   - $H_a : \mu_d < 0$ (Algorithm 1 is faster than Algorithm 2)

   Given the alternative hypothesis, our test is a left one-tailed test.

b. Determine the significance level, α. Common significance levels are 0.05 and 0.01.

$$\alpha = 0.05$$

c. Determine the sample size n

$$n = 46$$

d. Calculate the difference, d, between the runtime measurements of *Algorithm 1* and *Algorithm 2*. Since the alternative hypothesis is one-tailed, the order of the difference and sign distinction matter. It must be computed as

$$d_i = Algorithm1_i - Algorithm2_i$$

The resulting differences are:

	Algorithm 1	Algorithm2	Difference
1	138.0543456	318.9028852	-180.8485396
2	152.1015747	304.4452786	-152.3437039
3	99.52037024	334.5002985	-234.9799283
4	151.1789021	312.3957634	-161.2168613
5	114.6054552	274.3943777	-159.7889225
6	215.8103986	313.4142752	-97.6038766
7	112.1142957	402.1961331	-290.0818374
8	148.6598487	290.3641605	-141.7043118
9	149.6240342	291.8818815	-142.2578473
10	146.6564169	329.4828367	-182.8264198
11	297.2014318	274.9674318	22.234
12	152.1488256	270.3810124	-118.2321868
13	115.1343913	270.3849983	-155.250607
14	155.5894198	313.3392942	-157.7498744
15	405.4833677	339.8832877	65.60008
16	184.6442544	322.8176975	-138.1734431
17	173.6152842	321.3325841	-147.7172999
18	116.0371039	269.8869979	-153.849894
19	403.9426193	350.8998804	53.0427389
20	145.5398858	315.8995111	-170.3596253
21	132.1090143	316.3654263	-184.256412
22	147.5533206	310.8322659	-163.2789453
23	140.0441511	289.3754609	-149.3313098

(a) Run 1 - 23

	Algorithm 1	Algorithm2	Difference
24	148.5859799	273.8209417	-125.2349618
25	148.1272612	400.3326094	-252.2053482
26	159.1656756	320.9897439	-161.8240683
27	145.5891399	271.8324771	-126.2433372
28	158.6353071	270.3475635	-111.7122564
29	155.6163764	335.4172583	-179.8008819
30	126.6005828	311.4435296	-184.8429468
31	146.62609	327.1177959	-180.4917059
32	174.1569817	325.9297333	-151.7727516
33	139.6362522	380.4817438	-240.8454916
34	157.6228476	348.4538329	-190.8309853
35	169.646229	294.3979769	-124.7517479
36	135.5909698	335.9437046	-200.3527348
37	112.0925634	291.9122658	-179.8197024
38	180.6716633	309.0799878	-128.4083245
39	149.6013205	274.3581386	-124.7568181
40	153.1447635	267.8616531	-114.7168896
41	139.6289563	286.4070084	-146.7780521
42	204.6915617	272.3684094	-67.6768477
43	142.5906134	270.8283823	-128.2377689
44	167.6353343	353.4373901	-185.8020558
45	152.6202402	284.5009818	-131.8807416
46	126.5780046	287.8937998	-161.3157952

(b) Run 24 - 46

Table 3: Difference of runtime [s]

From Table 3 it can be observed that for 43 pairs of runs out of the 46 in total the time difference is negative, indicating that *Algorithm 1* was faster than *Algorithm 2* in 43/46 runs.

e. Calculate the mean of the difference, $\bar{d}$

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n}(Algorithm1_i - Algorithm2_i)$$

$$\bar{d} = \frac{1}{46}\sum_{i=1}^{46}(Algorithm1_i - Algorithm2_i)$$

$$\bar{d} = -146.5495052$$

The negative mean difference indicates that, on average, *Algorithm 1* is faster than *Algorithm 2* by 146.5495052 seconds.

Note: $\bar{d}$ was obtained manually and verified with the =AVERAGE function of Excel

f. Calculate the standard deviation of the difference, s_d

$$s_d = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left((Algorithm1_i - Algorithm_i) - \bar{d}\right)^2}$$

$$s_d = \sqrt{\frac{1}{46-1}\sum_{i=1}^{46}\left((Algorithm1_i - Algorithm_i) - (-146.5495052)\right)^2}$$

$$s_d = 65.56035238$$

Note: s_d was obtained manually and verified with the =STDEV.S function of Excel

g. Calculate the standard error of the mean of the difference, $SE(\bar{d})$

$$SE(\bar{d}) = \frac{s_d}{\sqrt{n}}$$

$$SE(\bar{d}) = \frac{65.56035238}{\sqrt{46}}$$

$$SE(\bar{d}) = 9.666346601$$

h. Determine the degrees of freedom

$$DoF = n - 1$$

$$DoF = 46 - 1$$

$$DoF = 45$$

i. Calculate the t-statistic, t

$$t = \frac{\bar{d}}{SE(\bar{d})}$$

$$t = \frac{-146.5495052}{9.666346601}$$

$$t = -15.16079562$$

j. Calculate the critical t-value for one-tailed test with significance level of α and degrees of freedom of DoF

$$t_{critical} = \pm 1.679427$$

Note: $t_{critical}$ was obtained with the =T.INV function of Excel

2. **Number of failures**

The appropriate statistical test to evaluate the number of failures is the Chi-Square Test, which allows to show a relationship between two categorical values ('Success' and 'Failure'). However, one of the conditions to perform this test is that no group should contain very few items (< 10). From the results, it was seen that *Algorithm 1* failed 1 time, while *Algorithm 2* failed 3 times. Since both samples contain very few items in the category 'Failure', a Chi-Square Test cannot be performed.

6 Conclusion

1. **Run Time**

From the statistical analysis, it can be seen that the critical t-value with a 95% confidence level with 45 degrees of freedom is approximately ± 1.6794, while the t-statistic resulted in -15.1608. In order to draw a conclusion, the absolute values of the t-statistic and the critical t-value must be compared:

For a left one-tailed test, if the value of the t-statistic is less than the negative value of the critical t-value, the Null hypothesis (H_0) is rejected

$$If \quad t < -t_{critical} \quad then \quad H_0 \quad is \quad rejected$$

Since the calculated t-value is significantly less than the negative critical t-value,

$$-15.1608 << -1.6794$$

we strongly reject the null hypothesis with 95% of confidence level. Therefore we can conclude that there is strong evidence to support the alternative hypothesis that: *Algorithm 1* is faster than *Algorithm 2*.

2. **Number of failures**

Since a Chi-Square Test could not be performed with the obtained measurements due to a very low number of items in one of the categories, it can be concluded that there is not enough data to accept nor reject the Null hypothesis.

Therefore, whether *Algorithm 1* has less failures than *Algorithm 2* or not is inconclusive.

In order to perform this categorical statistical test, more runs would be necessary until both categories reach the minimum number of items (i.e. until both algorithms have at least 10 failures). This however, remains as a suggestion for possible improvement but will not be implemented since already performing 50 runs on both algorithms was time expensive (> 6 hours).