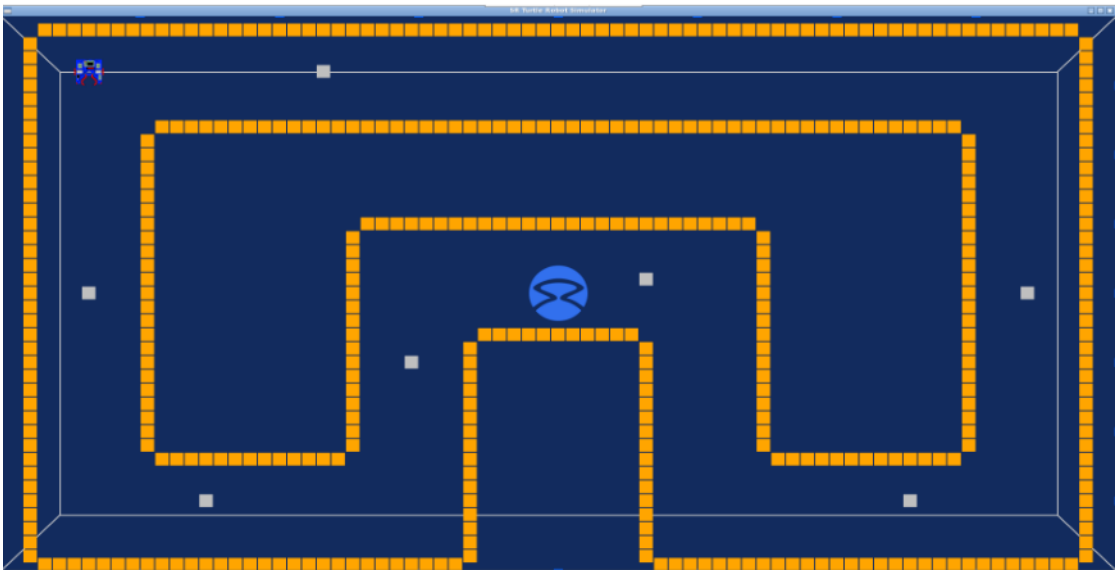
UNIVERSITÀ
DEGLI STUDI
DI GENOVA

RESEARCH TRACK 2

Statistical Study Report



Eléa PAPIN (s5248982)

12 may 2022

Contents

1 Introduction

In this report, I compare to solutions to a programming problem given as first assignment of the Research Track I class. To be precise, I compare the teacher solution to mine. This problem consists of a simulated environment in which a robot can move. The environment is made of corridors in which silver tokens can be found. For the assignment, we had to create a code to control the robot's movement in the corridors. The two codes have been written in the same language (python) and what I focus on is their ability to follow the following criteria:

- The robot should not collide with the walls
- The robot should drive through the corridor in one direction only
- When a silver token is on the robot's path, the robot should grab it and place it behind him

2 Conditions of data acquisition

In order to compare the two codes, I performed a statistical study. I first needed data sets of runs with both codes. I modified the environment slightly to place the silver tokens at random place within the corridors. I then ran the simulations 25 times for each of the codes and recorded the performances. The conditions in which all 50 simulations were run are the same, performed on the same computer with the same random placement generator.

2.1 Code to generate random tokens

Since there were 7 silver token in the original assignment, I decided to always place 7 tokens in the corridors. Since their positions are randomly generated, at first, tokens could appear in parts of the maps where the robot could not go, of within the corridors. To solve this issue, I defined zones which were inside the corridors and I drew random positions until the position was acceptable. I set a 10 try limit for each token, which means that I had to reject a few simulations that had one or two misplaced tokens.

Code Listing 1: Added part of code in sunny_side_up_arena.py

```
#Outside of the class
from random import random, randint

#zones which are all within the corridors
corridors = [ [-8.75, 8.75, -4.5,-3.25],
                [-8.75, -7.25, -4.5, 4.5],
                [-8.75, -1.75, 3.25, 4.5],
                [-3.25, -1.75, -1., 4.5],
                [-3.25, 3.25, -1.,0.5],
                [7.25, 8.75, -4.5, 4.5],
                [1.75, 8.75, 3.25, 4.5],
                [1.75, 3.25, 1., 4.5]
                ]

#will contain the seven tokens that are placed in the environment
placed_silver = []

def is_not_corridor(x,y):
    """Returns True if a token at (x,y) is not inside a corridor"""
    for cor in corridors:
        if cor[0]<=x<=cor[1] and cor[2]<=y<=cor[3]:
            return False
    return True
```

Code Listing 2: Added part of code in sunny_side_up_arena.py

```
def overlap(x,y):
    """Returns False if a token at (x,y) does not overlap with already
    placed silver tokens"""
    for tok in placed_silver:
        if abs(x-tok.location[0])<0.25 or abs(y-tok.location[1])<0.25:
            return True
    return False

#Within the SunnySideUpArena class, after the golden token placement
for i in range(7):
            x_temp = randint(-350, 350)/40
            y_temp = randint(-180, 190)/40
            tries= 0
            while (is_not_corridor(x_temp, y_temp)
            or overlap(x_temp,y_temp)) and tries <10:
                x_temp = randint(-350, 350)/40
                y_temp = randint(-180, 190)/40
                tries+=1

            token=SilverToken(self,count)
            token.location = (x_temp, y_temp)
            self.objects.append(token)
            placed_silver.append(token)
            count+1
```

2.2 Observed characteristics

For the all the runs, I observed the following elements:

- For each silver token placed, I noted whether the token was approached by the robot or not, and if it was, whether the robot successfully grabbed it and placed it behind him
- For each run, I counted the number of collisions between the robot and the walls
- For each run, I recorded whether the run ended in a U-turn, a full lap or if the robot ended up blocked in place
- Using a chronometer, I measured the time it took the robot to complete a run, whether the run ended at the end of the lap or was interrupted with a U-turn or a blocked robot

The raw data can be found on the file statistical_study_PAPIN.ods .

3 Non-parametric text

I will the use χ^2 law for characteritics not following a distribution. I have an overall size of 50 measurements and each group is of size 25>10. All measurements are independent and randomly obtained. The tests I will perform will all have a degree of freedom of 1. I will use a degree of significance of 5%.

3.1 Comparison of the ability to grab a nearby token

In all the simulations, the robot is driving to get close to the silver tokens he sees on its path. When the robot is close enough to one, it grabs the token. What I am comparing in this section is the robot's ability to grab a token if is passes nearby. Sometimes, the robot does not detect the closest silver token and drives past it. Sometimes, the token is too close to a wall and even if it is properly detected, the robot doesn't succeed to grab it. In the simulations, the tokens are marked grabbed successfully, not grabbed with robot nearby or not approached. It does not follow any distribution therefore I will do a non parametric test using the χ^2 law. I made the following hypothesis :

H_0: Both codes are equally good at grabbing a nearby token.
H_a: The professor's code is better at grabbing a nearby token.

	token successfully grabbed	token not grabbed with robot nearby
professor's algorithm	118	27
my algorithm	93	35
total	211	62
expectation	105.5	31

Table 1: Token grabbing ability

I have $\chi^2 = 2 * (13.5^2/105.5 + 4^2/31) = 4.49 > 3.84$.

Comparing the computed χ-value and the value of the χ^2 table, I can say that rejecting the null hypothesis, I would make an error with a probability inferior to 5%. Therefore, I can state that the teacher's code is better at grabbing the tokens that are in the robot's vicinity.

3.2 Comparison of the number of full laps

I ran all simulation for one lap, unless the robot made a U-turn or ended stuck in place. I want to see if the one of the two codes is better at making the robot drive a full lap. I made the following hypothesis :

H_0: Both codes are equally good at finishing a lap.
H_a: The professor's code is better at full lap.

	full lap	not full lap
professor's algorithm	12	13
my algorithm	11	14
total	23	27
expectation	11.5	13.5

Table 2: Ability to complete one lap

I have $\chi^2 = 0.08$. If I were to reject the null hypothesis, the χ-value indicates that I would make an error with a probability between 0.7 and 0.8. Therefore I cannot reject it. I cannot say that one code is better than the other at completing one lap.

3.3 Comparison of the U-turn risk

One of the codes' requirement was to make to robot drive in one direction only. But it turns out that both the professor's code and mine can cause the robot to make a U-turn and start driving in the wrong direction. I decided to stop a run a soon as such a U-turn happened. I am now comparing the risks of U-turn between both codes. I made the following hypothesis :

H_0: The robot has the same risk to do a U-turn with both codes.
H_a: The robot does more U-turns with my code.

	run without U-turn	run with a U-turn
professor's algorithm	18	7
my algorithm	11	14
total	29	21
expectation	14.5	10.5

Table 3: U-turn risk

I have $\chi^2 = 4.02 > 3.84$. The χ-value indicates that I can reject the null hypothesis with an error probability of 5%. The professor's code is less likely than mine to make the robot do a U-turn.

3.4 Comparison of collision avoidance ability

The codes are supposed to steer the robot away from the walls to avoid collisions with the golden tokens. During one run, the robot can collide with the walls without impairing its ability to drive so I just counted the number of collisions happening during each run. I want to see which code performs better to avoid collisions. I made the following hypothesis :

H_0: Both codes are equally good at avoiding collisions.
H_a: My code performs better on collision avoidance.

	run without collision	run with a least one collision
professor's algorithm	8	17
my algorithm	23	2
total	31	19
expectation	15.5	9.5

Table 4: Collision avoidance ability

I have $\chi^2 = 19.10$. The χ-value is really high, since 19.10>3.84, I can reject the null hypothesis. If I were to change the level of significance, I could even reject H_0 with an error probability inferior to 0.5%. My code is definitively better than the teacher's solution at avoiding collisions between the robot and the walls.

4 Parametric test

4.1 Speed comparison

To compare the speed of the robot with the two codes, I only used the data I had for runs in which the robot did a full lap of the environment. I chose to use a level of significance of 5% because I have a small sample size. I assumed that the time to perform such a task must follow a T-distribution (especially since the number of useful samples is low). Since I want to compare to data sets, I do a two sample T-test. I made the following hypothesis:

H_0: The robot finishes a lap just as fast with the two codes.
H_a: The robot is faster to finish one lap with my code.

	sample size	mean (s)	standard deviation (s)
professor's algorithm	12	268	25
my algorithm	11	242	31

Table 5: Speed comparison

$\hat{\sigma}_{pooled} = \frac{(N_1-1)*\sigma_1^2+(N_2-1)*\sigma_2^2}{N_1+N_2-2} = 785$
$\hat{\sigma}_{\bar{x}_1-\bar{x}_2} = \sqrt{\hat{\sigma}_{pooled} * (\frac{1}{N_1} + \frac{1}{N_2})} = 11.70$
$t_{\bar{x}_1-\bar{x}_2} = \frac{\bar{x}_1-\bar{x}_2}{\hat{\sigma}_{\bar{x}_1-\bar{x}_2}} = 2.22$

I have a degree of freedom of 21. With a 5% degree of significance, it the T-table gives a probability of 1.72. The computed t-value is greater than that (2,22<1.72) therefore I can reject the null hypothesis with an error probability of 5%. I can conclude that my code is faster that the teacher's code.

5 Conclusion

The previous statistical study shows that one code is not better than the other in general. In order to rank them, we have to established priority among the requirements. For example, if the robot was fragile, the collisions would need to be avoided at all costs and my code would be safer. But if the robot was robust or the environment was forgiving, then the teacher's code would be better as it is better at the grabbing task. Without an evaluation grid, I can just state that both codes have strengths and weaknesses.