Robotics Lab 1: Open Loop

Authors: Stuart Harley, Kiel Dowdle
Date: 9/9/2021

Abstract

This lab is used as our introduction to Lego Mindstorm robots. We experiment with motor functions and timing in order to move the robot along specified paths. Our robot control in this lab is open-loop which means that the robot system receives no feedback or information about the environment. Due to this, we found that there are many causes of error in the execution of our instructions caused by both internal and external factors. Because this is an open-loop system, the robot has no way to correct any generated error. Therefore, the more actions that are taken in an open-loop system, the more error will be accumulated.

We perform 3 tasks in the lab. The first is to move a specified distance in a straight line. We found that a slower speed made the robot move a more accurate distance. The type of ground the robot moved on also impacted the accuracy of the movement. The second task is to rotate the robot 90 degrees. We derived the equation but due to several sources of error, the robot typically over-rotated by several degrees. The third task was to move the robot along 3 different specified paths made up of straight sections and turns. While we were able to find a speed for straight paths that was statistically accurate in task 1, we were not able to make our robot turn 90 degrees accurately. Because of this, the paths that included more turns ended up being more inaccurate.

Methods

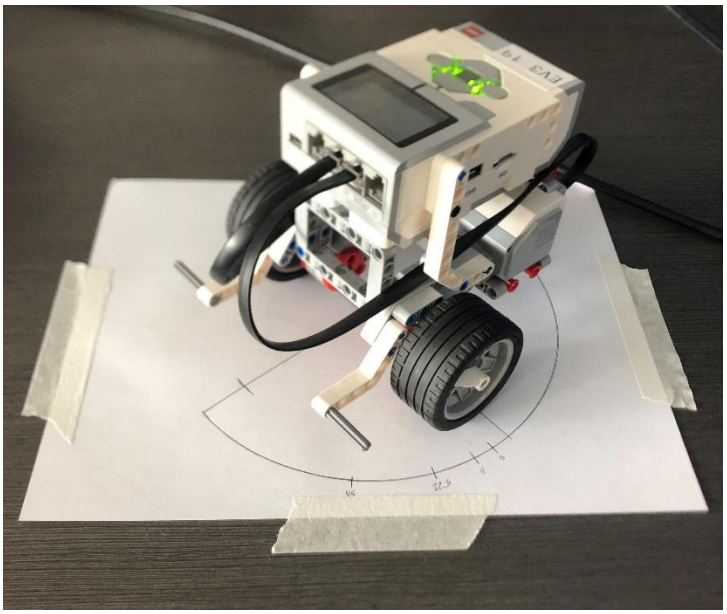
For task 1 (Forward Movement), we derived the equation that specifies how far the robot moves based on the duration of the run command, the speed, and the circumference of the wheels. We measured the radius of the wheels to be 2.8cm. The circumference calculates to be 17.59cm using the formula circumference = 2 * pi * radius. Our speed we put in terms of rotations per minutes, so our formula for forward distance is d = min * rpm * 17.59.

For measuring the distance the robot moved in task 1, a ruler was placed next to the robot and the robot was placed with the center of the wheel at 0cm. The robot moved forward until stopping and the distance at the center of the wheel was recorded. Example picture below.

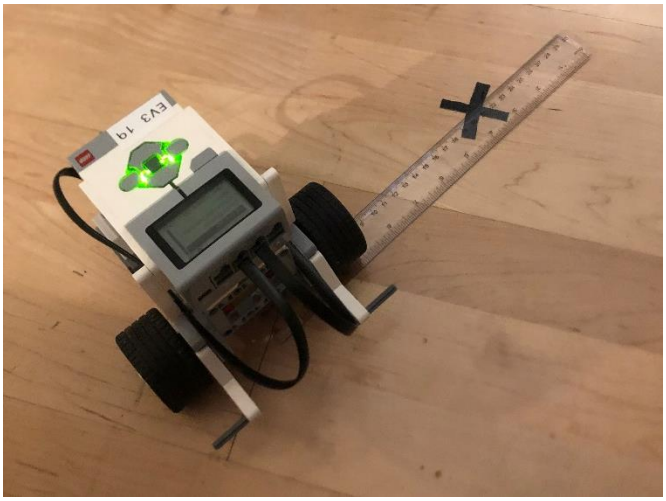
For task 2 (Rotation), we derived the equation that specifies how far the robot needs to spin its wheels in opposite directions to rotate the robot by a specified amount. Our equation was based on the duration of the run command, the speed, the radius of the axle, and the circumference of the wheels. We measured the

radius of the axle to be 5.9cm. Therefore, the circumference of the turning radius of the robot is 37.07cm. Our speed again is in terms of rpm, and the terms for rotation are the number of full rotations completed. Therefore, our formula is rotations = min * rpm * 17.59 / 37.07. This equation assumes that one wheel is moving in the positive direction and the other is moving the negative direction.

For measuring the actual amount the robot rotated in task 2, a sheet of paper was marked with a circle and starting locations for the wheels. The robot rotated until stopping and the actual degrees rotated were recorded. Example picture below.



For task 3 (Path traversal) we used the formulas we derived in tasks 1 and 2 to traverse 3 paths. The goal was for the robot to return to its starting location after traversing the path. The distance between the actual start and end points was recorded. Example picture below.



Results

After completing each task and recording the results we performed a one-sided T-test to determine if the robot moved as we were expecting. The null hypothesis for each test was that the robot moves matched the expected value. The alternative hypothesis is that the robot's moves do not match the expected value. In this case we are rejecting the null hypothesis when the p-value of the test is less than or equal to 0.05 ($p <= 0.05$).

If we accept the null hypothesis the p-value gives us the probability of the measured mean appearing in a normal distribution of measurements.

Each task is associated with a table showing the result of each trial and the corresponding p-value of the one-sided T-test. The statistical test was performed on Microsoft Excel using the T. DIST.RT function (Right tail). This means when the mean is less than the hypothesized mean we need to take the absolute of the of the difference between the mean and the hypothesized mean.

Task 1 (Forward Movement)

For this task we calculated three sets of duration and speed (rpm) parameters that, according to our equation from above, should make the robot move 25cm. We tested each combination 5 times on hardwood, and 5 times on carpet. Results are shown in Table 1. P-values for each trial were calculated using a one-sided t-test. Our null hypothesis was that the robot moved the expected distance of 25cm. The trial when the robot was set to 8.53 rpms and was ran for 0.1667 minutes on Hardwood had a p-value of 0.3971 and therefore was the only setup where the null hypothesis was not rejected. Every other trial rejected the null hypothesis making the alternate hypothesis stating that the robot did not move the expected distance of 25cm true.

	rpm=60, min=.0237		rpm=8.53, min=.1667		rpm=130, min=.0109	
	Hardwood 1 [cm]	Carpet 1 [cm]	Hardwood 2 [cm]	Carpet 2 [cm]	Hardwood 3 [cm]	Carpet 3 [cm]
Trial 1	30.3	25.5	24.9	23.1	40.1	32.8
Trial 2	30.6	26.9	24.9	22.9	39.8	30.8
Trial 3	30.1	25.9	25.0	23.3	40.0	31.8
Trial 4	30.2	26.6	24.8	23.3	39.1	30.9
Trial 5	31.4	26.4	25.6	23.0	40.4	32.5
Mean	30.52	26.26	25.04	23.12	39.88	31.76
P-value	9.798E-06	0.0036	0.3971	9.719E-06	1.372E-07	3.800E-05

Table 1: Task 1 measurements and one-sided t-test results

Following below (Fig 1, 2, 3) is three box plots visually describing the distribution of measurements for each of the experimental setups.

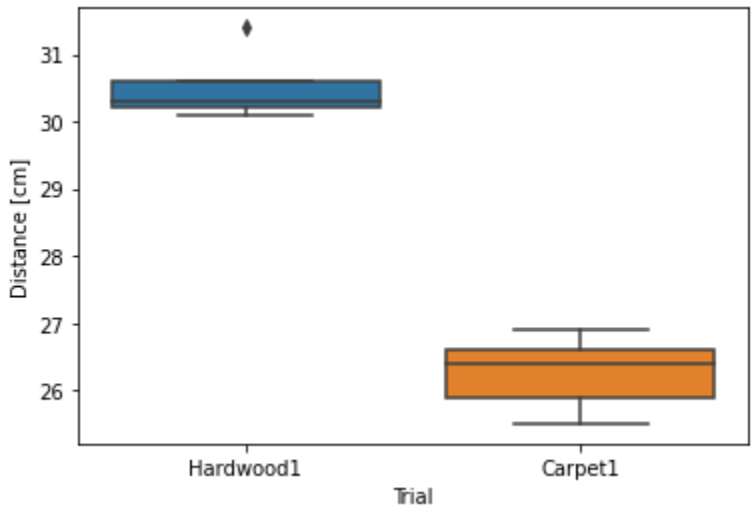


Figure 1: Box plot describing the distribution for the first experimental setup (rpm=60, min=.0237)

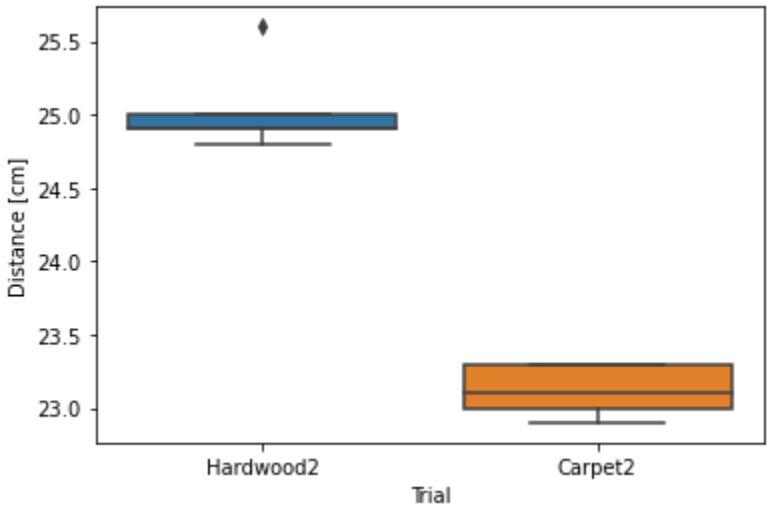


Figure 2: Box plot describing the distribution for the second experimental setup (rpm=8.53, min=.1667).

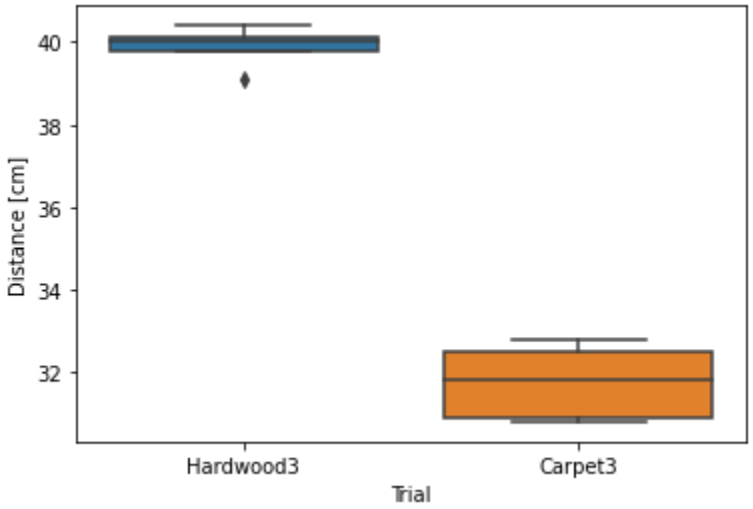


Figure 3:Box plot describing the distribution for the third experimental setup (rpm=130, min=.0109).

Task 2 (Rotation)

For the next task we used our derived rotation formula to perform what should be a 90-degree clockwise rotation and a 90-degree counterclockwise rotation of our robot. We used the most accurate rpm from the previous task which was 8.53 rpm. We tested each rotation 5 times. The results for this task can be found below in Table 2. P-values for each trial were calculated using a one-sided t-test. Our null hypothesis was that the robot rotated the expected 90 degrees. Moving counterclockwise seemed to be more accurate and was backed up by a p-value of 0.1311 meaning the null hypothesis was not rejected. Moving clockwise in comparison had a p-value of 0.0027 therefore we were able to reject the null hypothesis so the alternate hypothesis that the robot did not rotate the expected 90 degrees was supported.

	90 Clockwise [deg]	90 Counterclockwise [deg]
Trial 1	94.2	92.7
Trial 2	93.6	84.3
Trial 3	93.2	93.9
Trial 4	97.7	83.7
Trial 5	92.8	88.2
Mean	94.3	88.56
P-value	0.0027	0.1311

Table 2: Task 2 measurements and one-sided t-test results.

Figure 4 below contains a box plot providing a visualization of the distribution of measurements.

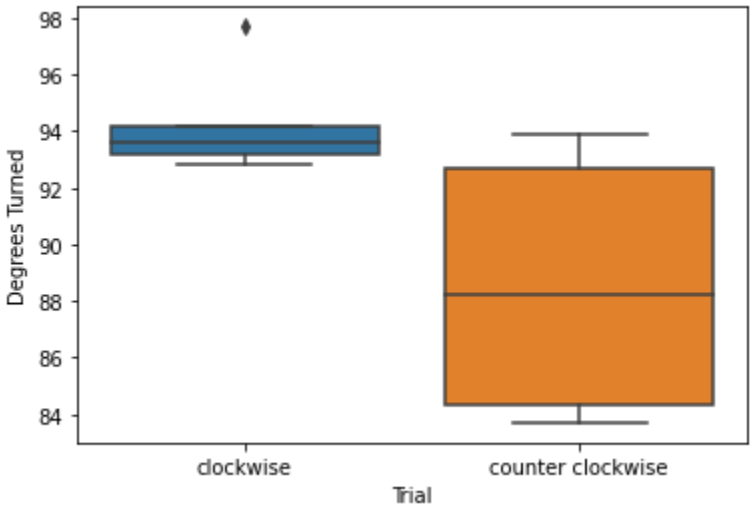


Figure 4: Box plot describing the results for rotating the robot clockwise and counterclockwise

Task 3 (Path traversal)

For the final task we tested each path 5 times. The results for this task can be found below in Table 3. P-values for each trial were calculated using a one-sided t-test. Our null hypothesis was that the robot followed the path and returned to the starting location. For all three paths we rejected the null hypothesis and accepted the alternate hypothesis that the robot did not follow the path and return to the starting location.

Distance between start and end point of path			
	Path 1 (Line) [cm]	Path 2 (Flag) [cm]	Path 3 (Hammer) [cm]
Trial 1	18.5	22.2	20.7
Trial 2	19.8	15.9	20.5
Trial 3	17.8	19.5	16.4
Trial 4	17.4	25.3	18.8
Trial 5	16.8	17.8	18.3
Mean	18.06	20.14	18.94
P-value	1.267E-06	8.433E-05	5.693E-06

Table 3: Task 3 measurements and one-sided t-test results.

Figure 5 below contains a box plot providing a visualization of the distribution of measurements between the starting point and ending point of each path.

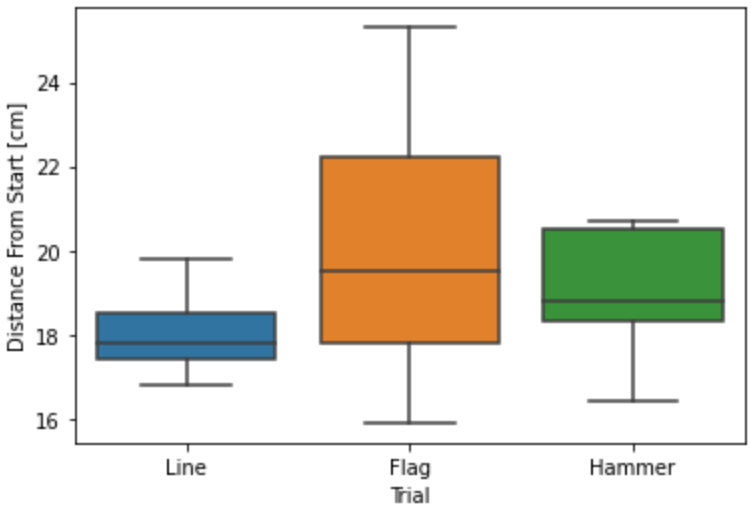


Figure 5: Box plot describing the results for the robot following the three different paths.

Shown below is a screenshot of a video taken of the robot completing path 3.

Discussion

1. For the forward movement, the robot was more accurate when moving at a slower speed. The higher the speed got, the further the robot went past the intended distance.

2. The floor type affected the distance that the robot moved. The carpeted floor caused the robot to move shorter distances because there was more friction between the wheels and the floor. The hardwood floors had less friction with the wheels, so the robot was able to move further.

3. One of the factors that caused differences between the predicted and actual distance was how the robot stop method functions. It does not apply any brakes and immediately stop the robot. Instead, it cuts the power to the motor, therefore the robot still has its momentum, so it continues to move forward until friction brings it to a full stop.

   Another factor is that any minor timing issues with delays in code execution or inaccuracies in the time.sleep function cause the robot to move forward further. There were also errors due to the inability to reset the robot to the exact same location for every run. Also, there was error in our measurements since we were using a simple ruler. These inaccuracies caused our derived equations to be inaccurate.

4. Rotating clockwise appeared to rotate further than rotating counterclockwise. It is not clear why this tended to be the case. It is possible that the order in which the motors were started and stopped could have caused this.

5. The path with the smallest end location error was path 1. For task 3, most of our error came from the inaccuracies of the turning since the robot tended to turn further than the desired amount. Since path 1 involved only 1 turn, the amount of turning error was less than the other paths.

Supplementary Material

Task 1 code:

```
#!/usr/bin/env pybricks-micropython
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                 InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import Port, Stop, Direction, Button, Color
from pybricks.tools import wait, StopWatch, DataLog
from pybricks.robotics import DriveBase
from pybricks.media.ev3dev import SoundFile, ImageFile

import time

def move(rpm, min):
    deg_per_sec = rpm / 60 * 360
    sec = min * 60
    motor1.run(deg_per_sec)
    motor2.run(deg_per_sec)
    time.sleep(sec)
    motor1.stop
    motor2.stop

ev3 = EV3Brick()
motor1 = Motor(Port.B)
motor2 = Motor(Port.C)

# Tests for 25cm
move(60, .0237)
# move(8.53, .1666)
# move(130, .0109)
```

Task 2 code:

```
#!/usr/bin/env pybricks-micropython
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                 InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import Port, Stop, Direction, Button, Color
from pybricks.tools import wait, StopWatch, DataLog
from pybricks.robotics import DriveBase
from pybricks.media.ev3dev import SoundFile, ImageFile

import time

def rotateCW(full_rotations, rpm=8.53, wheel_c=17.59, axle_c=37.07):
    # best rpm from past task = 8.53
    deg_per_sec = rpm / 60 * 360
    sec = full_rotations / (rpm * wheel_c / axle_c) * 60
    motor1.run(deg_per_sec)
    motor2.run(-deg_per_sec)
    time.sleep(sec)
    motor1.stop
    motor2.stop

def rotateCCW(full_rotations, rpm=8.53, wheel_c=17.59, axle_c=37.07):
    deg_per_sec = rpm / 60 * 360
    sec = full_rotations / (rpm * wheel_c / axle_c) * 60
    motor1.run(-deg_per_sec)
    motor2.run(deg_per_sec)
    time.sleep(sec)
    motor1.stop
    motor2.stop

ev3 = EV3Brick()
motor1 = Motor(Port.B)
motor2 = Motor(Port.C)
rotateCW(.25)
# rotateCCW(.25)
```

Task 3 code:

```
#!/usr/bin/env pybricks-micropython
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                 InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import Port, Stop, Direction, Button, Color
from pybricks.tools import wait, StopWatch, DataLog
from pybricks.robotics import DriveBase
from pybricks.media.ev3dev import SoundFile, ImageFile

import time

# Default values make the robot move 25cm straight
def move(rpm=8.53, min=.1666):
    deg_per_sec = rpm / 60 * 360
    sec = min * 60
    motor1.run(deg_per_sec)
    motor2.run(deg_per_sec)
    time.sleep(sec)
    motor1.stop
    motor2.stop

def rotateCW(full_rotations, rpm=8.53, wheel_c=17.59, axle_c=37.07):
    # best rpm from past task = 8.53
    deg_per_sec = rpm / 60 * 360
    sec = full_rotations / (rpm * wheel_c / axle_c) * 60
    motor1.run(deg_per_sec)
    motor2.run(-deg_per_sec)
    time.sleep(sec)
    motor1.stop
    motor2.stop

def rotateCCW(full_rotations, rpm=8.53, wheel_c=17.59, axle_c=37.07):
    deg_per_sec = rpm / 60 * 360
    sec = full_rotations / (rpm * wheel_c / axle_c) * 60
    motor1.run(-deg_per_sec)
    motor2.run(deg_per_sec)
    time.sleep(sec)
    motor1.stop
    motor2.stop

ev3 = EV3Brick()
motor1 = Motor(Port.B)
motor2 = Motor(Port.C)

# Path 1 (Straight Line)
move()
time.sleep(.5)
move()
time.sleep(.5)
rotateCW(.5)
time.sleep(.5)
move()
time.sleep(.5)
move()
```

```
# Path 2 (Flag)
move()
time.sleep(.5)
move()
time.sleep(.5)
rotateCW(.25)
time.sleep(.5)
move()
time.sleep(.5)
rotateCW(.25)
time.sleep(.5)
move()
time.sleep(.5)
rotateCW(.25)
time.sleep(.5)
move()
time.sleep(.5)
rotateCCW(.25)
time.sleep(.5)
move()

# Path 3 (Hammer)
move()
time.sleep(.5)
rotateCCW(.25)
time.sleep(.5)
move()
time.sleep(.5)
rotateCW(.25)
time.sleep(.5)
move()
time.sleep(.5)
rotateCW(.25)
time.sleep(.5)
move()
time.sleep(.5)
move()
time.sleep(.5)
rotateCW(.25)
time.sleep(.5)
move()
time.sleep(.5)
rotateCW(.25)
time.sleep(.5)
move()
time.sleep(.5)
rotateCCW(.25)
time.sleep(.5)
move()
```

Excel sheet for statistical tests:

Task 1 results 1 tailed t test						
	rpm=60, min=.0237		rpm=8.53, min=.1666		rpm=130, min=.0109	
	Hardwood 1	Carpet 1	Hardwood 2	Carpet 2	Hardwood 3	Carpet 3
Trial 1	30.3	25.5	24.9	23.1	40.1	32.8
Trial 2	30.6	26.9	24.9	22.9	39.8	30.8
Trial 3	30.1	25.9	25	23.3	40	31.8
Trial 4	30.2	26.6	24.8	23.3	39.1	30.9
Trial 5	31.4	26.4	25.6	23	40.4	32.5
mean	30.52	26.26	25.04	23.12	39.88	31.76
std dev	0.526307895	0.559464029	0.320936131	0.178885438	0.486826458	0.907193474
n	5	5	5	5	5	5
std err	0.235372046	0.25019992	0.143527001	0.08	0.217715411	0.405709256
df	4	4	4	4	4	4
hyp mean	25	25	25	25	25	25
t-stat	23.45223273	5.035972832	0.278693206	23.5	68.34610357	16.66217842
p-value	9.79803E-06	0.003651362	0.397147371	9.71908E-06	1.37292E-07	3.80047E-05

Task 2 results		
	clockwise	counter clockwise
Trial 1	94.2	92.7
Trial 2	93.6	84.3
Trial 3	93.2	93.9
Trial 4	97.7	83.7
Trial 5	92.8	88.2
mean	94.3	88.56
std dev	1.761817244	4.184542986
n	5	5
std err	0.787908624	1.871384514
df	4	4
hyp mean	90	91
t-stat	5.457485638	1.303847489
p-value	0.002739663	0.13112928

Distance between start and end point of path (cm)			
	Path 1 (Line)	Path 2 (Flag)	Path 3 (Hammer)
Trial 1	18.5	22.2	20.7
Trial 2	19.8	15.9	20.5
Trial 3	17.8	19.5	16.4
Trial 4	17.4	25.3	18.8
Trial 5	16.8	17.8	18.3
mean	18.06	20.14	18.94
std dev	1.030727898	3.308534419	1.575563391
n	5	5	5
std err	0.460955529	1.479621573	0.704613369
df	4	4	4
hyp mean	0	0	0
t-stat	39.17948447	13.61158851	26.87998955
p-value	1.26766E-06	8.43371E-05	5.69388E-06