Bài thực hành môn UDCNM năm 2017

Bộ môn: KHMT

**Bài thực hành tuần 1-5: Làm quen với ngôn ngữ Python**

**Bài thực hành số 1.**

**Nội dung:**

* Cài đặt chương trình soạn thảo đơn giản cho Python
* Thực hành viết chương trình đơn giản trên Python
* Thực hành các toán tử trong ngôn ngữ lập trình Python

**Yêu cầu:**

Công cụ soạn thảo Python

**Chi tiết:**

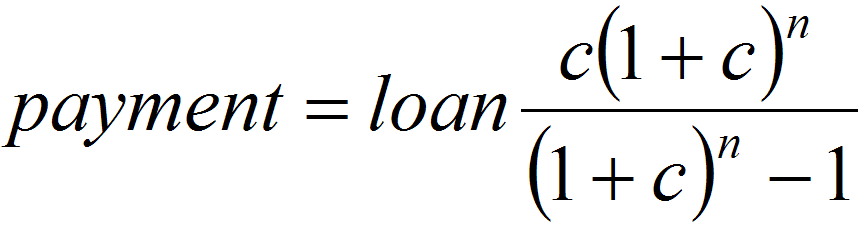
1. Write a program that asks the user about textbook prices and reports how overpriced the textbooks are. (You may wish to read this number from the user with the input command. You can round numbers with the round command.) You should match the following output

* How much do you want to pay for each textbook? **50**
* How much did the average textbook actually cost? **80**
* How many books did you buy? **5**
* Each book is overpriced by $30 ( 60% )
* You got ripped off by $150 total

You may assume that the user enters a positive integer for each input value above.

1. If you ever find yourself buying a house, you'll want to know what your monthly payment for the loan is going to be. Write a complete program that asks for information about a loan and prints the monthly payment.

The formula for computing monthly mortgage payments involves the loan amount, the total number of months involved (a value we call *n*) and the monthly interest rate (a value we call *c*). The payment formula is given by the following equation:



An example run of your program might produce the following output (user input is underlined):

* This program computes monthly loan payments.
* Loan amount? **275000**
* Number of years? **30**
* Interest rate? **6.75**
* Your payment is $1783

Note that the numbers being read as input don't match the way they should be used in the formula. The term of the loan is read as years instead of months. The interest rate percentage is being read in as a yearly rate instead of a monthly rate and as a whole number rather than a true percentage, so the value 6.75 should actually become 0.005625 by dividing it by 12 and by 100. You will have to make these conversions as part of your program.

1. Write a program that counts a number's factors and determines whether the number is prime.

* What is your favorite number? **24**
* 24 has 8 factors
* 24 is not prime
* What is your favorite number? **31**
* 31 has 2 factors
* 31 is prime

Hint: To count the factors of some integer *n*, use a loop that tests every integer less than *n* to see whether *n* is divisible by it. (How do you know whether one integer is divisible by another?)

**Bài thực hành số 2.**

**Nội dung:**

* Thực hành các nhập xuất chuỗi, số trên Python
* Thực hành các câu lệnh rẽ nhánh, lặp…trên Python

**Yêu cầu:**

Công cụ soạn thảo Python

**Chi tiết:**

You may not know that credit card numbers contain several pieces of information for performing validity tests. For example, Visa card numbers always begin with 4, and a valid Visa card number also passes a digit-sum test known as the Luhn checksum algorithm. Luhn's algorithm states that if you sum the digits of the number in a certain way, the total sum must be a multiple of 10 for a valid Visa number. Systems that accept credit cards perform a Luhn test before contacting the credit card company for final verification. This lets the company block fake or mistyped credit card numbers.

The algorithm for summing the digits is the following. Consider each digit of the credit card to have a zero-based index: the first is at index 0, and the last is at index 15. Start from the rightmost digit and process each digit one at a time. For digits at even-numbered indexes (the 14th digit, 12th digit, etc.), simply add that digit to the cumulative sum. For digits at odd-numbered indexes (the 15th, 13th, etc), double the digit's value, then if that doubled value is less than 10, add it to the sum. If the doubled number is 10 or greater, add each of its digits separately into the sum.

The following pseudocode describes the Luhn algorithm to sum the digits:

sum = 0.

for each digit of credit card number, starting from right,

if the digit's index is even,

add the digit to sum.

else,

double the digit's value.

if the doubled value is less than 10,

add the doubled value to sum.

else,

split the doubled value into its two digits.

add the first digit to sum.

add the second digit to sum.

4111111111111111 and 4408041254369873 are example credit card numbers that pass the Luhn algorithm. The following figure shows the algorithm summing the latter number in detail. Notice how digits at even indexes are doubled and potentially split into two digits if they exceed 10 when doubled. For example, the number 7 at index 8 which is doubled to 14 which split to make 1+4.

|  |
| --- |
| An example checksum using the Luhn algorithm. |
| CC # 4408 0412 5436 9873  4 4 0 8 0 4 1 2 7 4 3 6 9 8 5 3  Scale \*2 \*2 \*2 \*2 \*2 \*2 \*2 \*2  --------------------------------------------------------------------  8 4 0 8 0 4 2 2 14 4 6 6 18 8 10 3  Sum = 8 + 4 + 0 + 8 + 0 + 4 + 2 + 2 + 1+4 + 4 + 6 + 6 + 1+8 + 8 + 1+0 + 3  = 70  70 is divisible by 10, therefore this card number is valid. |

Write a program where the user can type in a credit card number and receive a message stating whether the number was valid.

**Bài thực hành số 3.**

**Nội dung:**

* Thực hành các xử lý file (đọc, thống kê) trên Python
* Thao tác trực tiếp File dữ liệu (ghi, ghi file cấu trúc) trên Python

**Yêu cầu:**

Công cụ soạn thảo Python

**Chi tiết:**

Suppose we have this hours.txt data:

* 123 Suzy 9.5 8.1 7.6 3.1 3.2
* 456 Brad 7.0 9.6 6.5 4.9 8.8
* 789 Jenn 8.0 8.0 8.0 8.0 7.5

Compute each worker's total hours and hours/day.

* Assume each worker works exactly five days.
* Suzy ID 123 worked 31.4 hours: 6.3 / day
* Brad ID 456 worked 36.8 hours: 7.36 / day
* Jenn ID 789 worked 39.5 hours: 7.9 / day

input = open("hours.txt")

for line in input:

id, name, mon, tue, wed, thu, fri = line.split()

**# cumulative sum of this employee's hours**

hours = float(mon) + float(tue) + float(wed) + \

float(thu) + float(fri)

print(name, "ID", id, "worked", \

hours, "hours: ", hours/5, "/ day"

**Bài thực hành số 4.**

**Nội dung:**

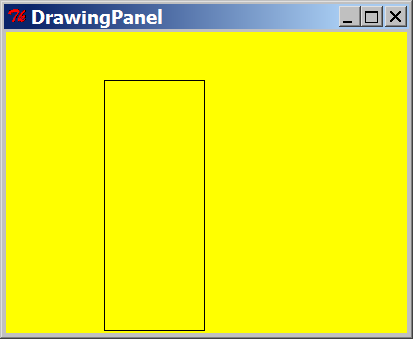
* Thực hành các hàm đồ họa cơ bản trên Python
* Cách trình diễn các đối tượng đồ họa cơ bản

**Yêu cầu:**

Công cụ soạn thảo Python

**Chi tiết:**

Writing code below:

****from drawingpanel import \*

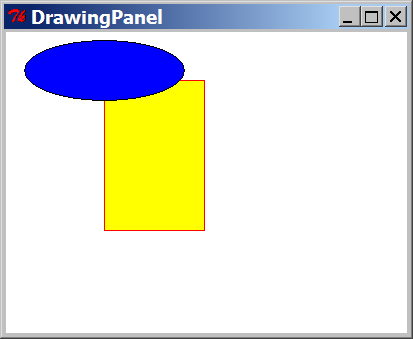
panel = DrawingPanel(400, 300)

panel.set\_background("yellow")

panel.canvas.create\_rectangle(100, 50, 200, 300)

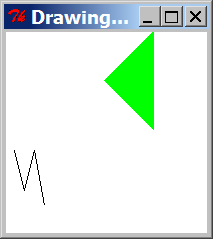
from drawingpanel import \*

panel = DrawingPanel(400, 300)

panel.canvas.create\_rectangle(100, 50, 200, 200, outline="red", fill="yellow")

panel.canvas.create\_oval(20, 10, 180, 70, fill="blue")

from drawingpanel import \*

panel = DrawingPanel(200, 200)

panel.canvas.create\_polygon(100, 50, 150, 0,

150, 100, fill="green")

panel.canvas.create\_line(10, 120, 20, 160,

30, 120, 40, 175)

DrawingPanel panel = new DrawingPanel(200, 200);

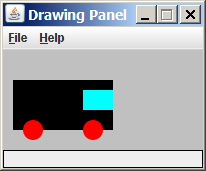
panel.setBackground(Color.LIGHT\_GRAY);

Graphics g = panel.getGraphics();

g.setColor(Color.BLACK); // body

g.fillRect(10, 30, 100, 50);

g.setColor(Color.RED); // wheels

****g.fillOval(20, 70, 20, 20);

g.fillOval(80, 70, 20, 20);

g.setColor(Color.CYAN); // windshield

g.fillRect(80, 40, 30, 20);

**Bài thực hành số 5.**

**Nội dung:**

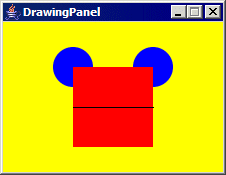
* Thực hành các hàm đồ họa trên Python
* Sử dụng các cấu trúc đồ họa phức tạp

**Yêu cầu:**

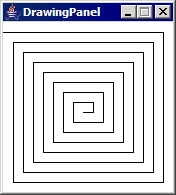
Công cụ soạn thảo Python

**Chi tiết:**

1. Write a program that draws the following figure:

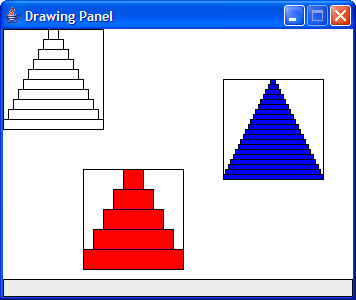


1. Write a program that draws the following figure:



Part of the challenge is using loops to reduce the redundancy of the figure.

1. Write a program that draws the following figure:



Part of the challenge is using loops to reduce the redundancy of the figure.

**Bài thực hành từ tuần 6-9 (Machine learning & Cloud Computing).**

**Bài thực hành số 6**

**Cài đặt môi trường triển khai ứng dụng machine-learning trên windows azure.**

**Nội dung:**

* Cài đặt được bộ free Microsoft Azure ML workspace
* Cấu hình được Setup Azure ML workspace với existing Azure subscription
* Cài đặt Python cho môi trường phát triển
* Cài đăth công cụ R phần thống kê

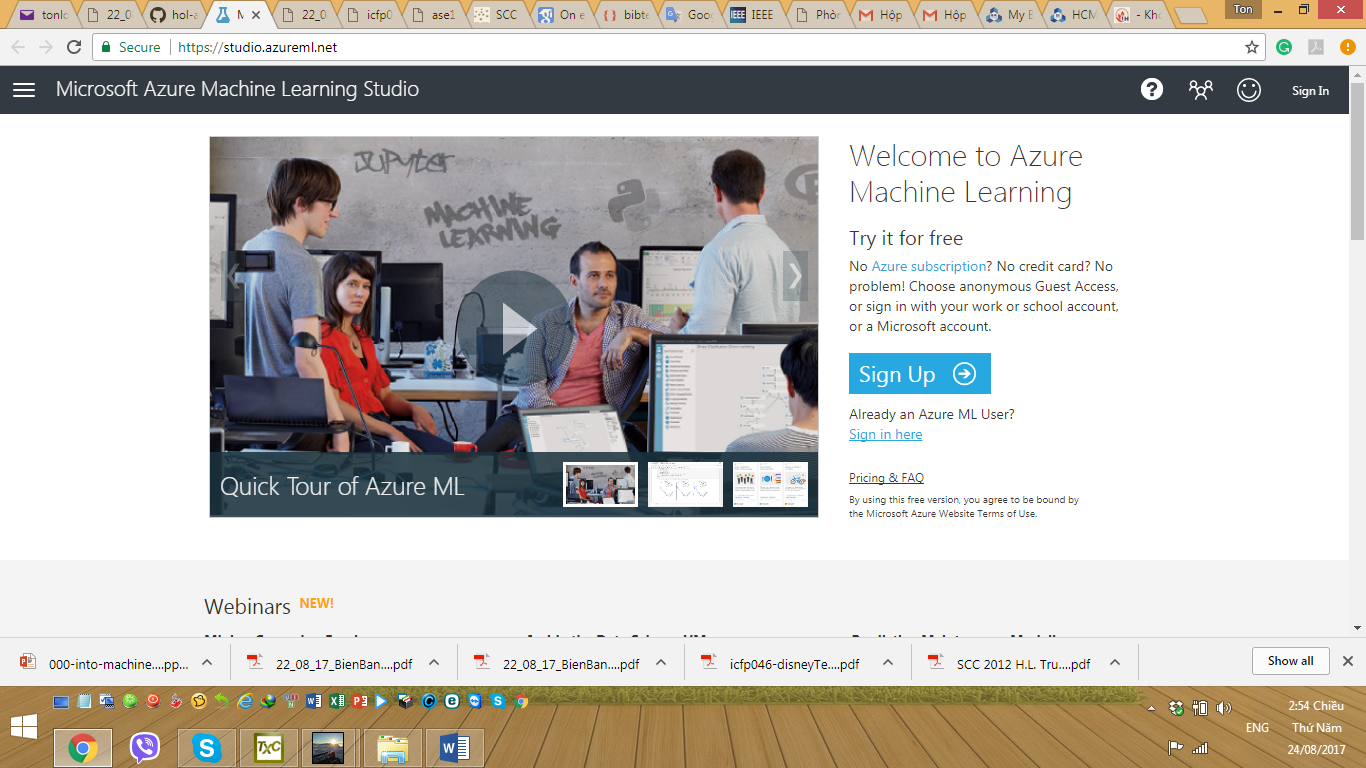
**Yêu cầu:**

**Chuẩn bị tài nguyên (phần mềm, internet, máy tính)**

**Chi tiết:**

**Đăng ký tài khoản free tại.**

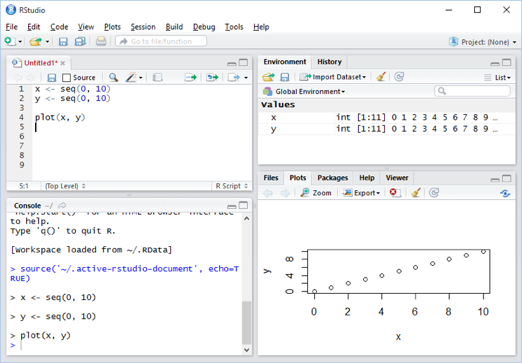
* Microsoft Account (signup at [https://signup.live.com](https://signup.live.com/))



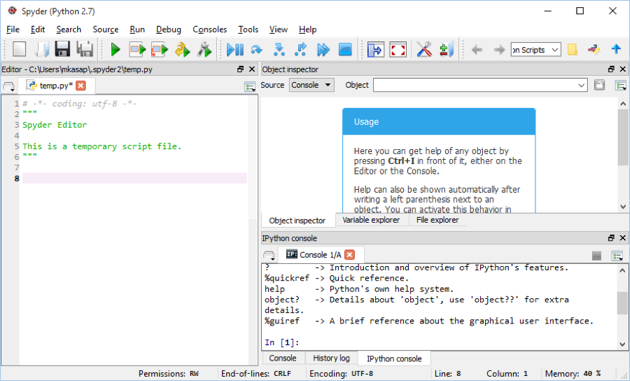
**Cài đặt phần mềm R\_Studio**

[https://cran.rstudio.com](https://cran.rstudio.com/) hoặc

<https://www.rstudio.com/products/rstudio/download/>



**Cài đặt Python từ** <https://www.continuum.io/downloads>



# Bài thực hành số 7: Giới thiệu R và xử lý dữ liệu với Python

**Nội dung:**

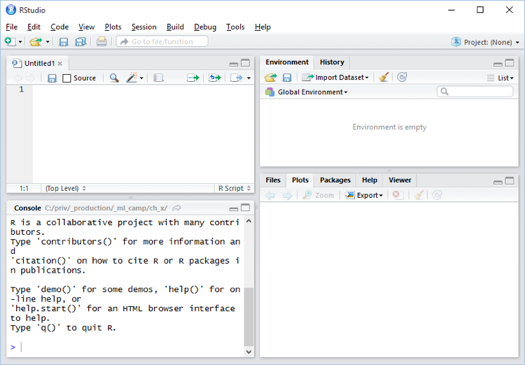
* Sử dụng được công cụ R vào ứng dụng thống kê trên dữ liệu
* Dùng được các hàm thống kê trong Python

**Yêu cầu:**

Đã cài đặt các công cụ ở buổi 6

**Chi tiết:**

Chạy Rstudio.



Gõ Script

# Generate a numeric series from 1 to 30 and assign it to variable x

x <- seq(1, 30)

# Create copy of x as variable y

y <- x

# Generate 30 uniform distributed random number each ranges between -1 to 1

noise <- runif(30, -1, 1)

# Create variable ywnoise as in Excel

ywnoise <- y + noise \* 2

# Plot values of x and ywnoise without labels

plot(x, ywnoise, xlab = NA, ylab = NA)

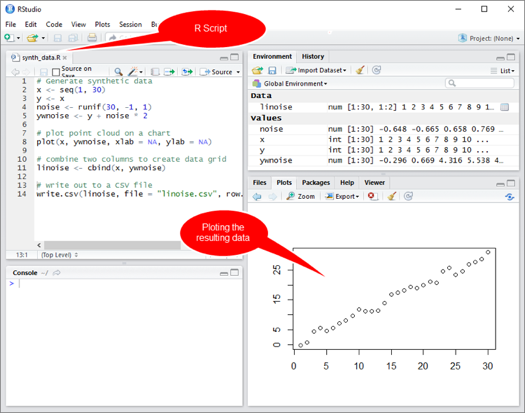
# Combine columns x and ywnoise to create two column grid named linoise

linoise <- cbind(x, ywnoise)

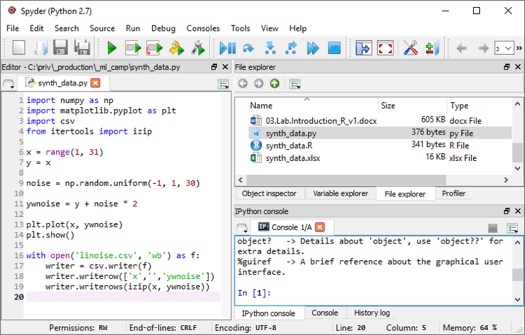
# Save the variable linoise as a CSV file on the local disk

write.csv(linoise, file = "linoise.csv", row.names = FALSE)

Kiểm tra output



Chạy IDE Python



Gõ Script

# Import required libraries to be used. i.e. csv for csv file output, pyplot for plotting etc.

import numpy as np

import matplotlib.pyplot as plt

import csv

from itertools import izip

# Generate identical x and y variables with numeric series from 1 to 30

x = range(1, 31)

y = x

# Generate 30 uniform distributed random number each range between -1 to 1

noise = np.random.uniform(-1, 1, 30)

# Create noisy y values with magnitude of 2

ywnoise = y + noise \* 2

# Plot the resulting x and ywnoise data

plt.plot(x, ywnoise)

plt.show()

# Write out the resulting data as a CSV file. Be carefull about the tab indentation which is important for Python.Write out the resulting data as a CSV file. Be carefull about the tab indentation which is important for Python.

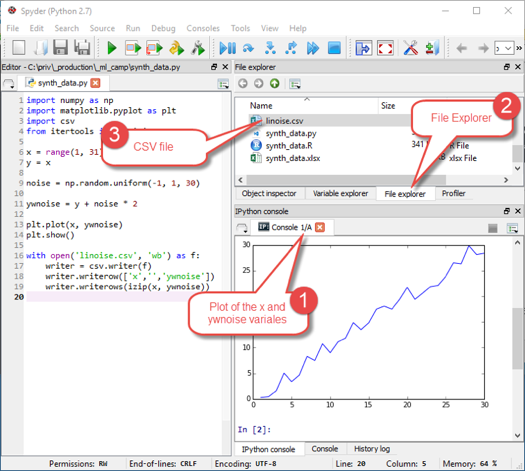
with open('linoise.csv', 'wb') as f:

writer = csv.writer(f)

writer.writerow(['x', 'ywnoise'])

writer.writerows(izip(x, ywnoise))

Kiểm tra kết quả



**Bài thực hành số 8: Phân tích dữ liệu trên AzureML**

**Nội dung:**

* Phân tích dữ liệu và trực quan hóa dữ liệu trên AzureML
* Rút trích dữ liệu

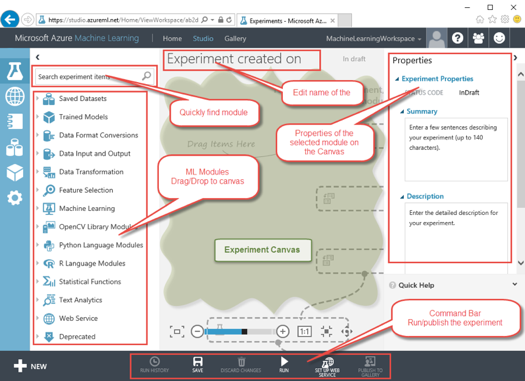
**Yêu cầu:**

Đã cài đặt các công cụ ở buổi 6

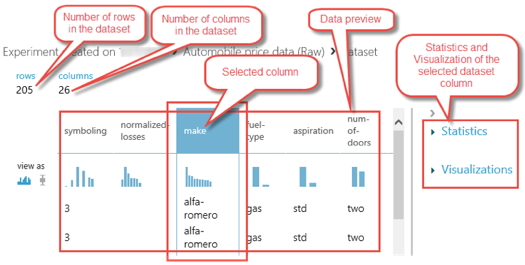
**Chi tiết:**

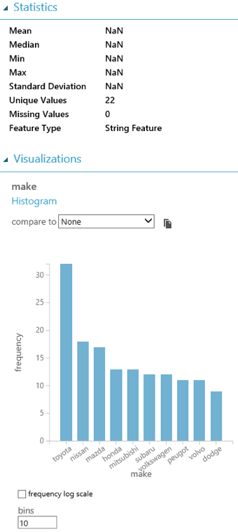
## Chạy AzureML Experiment.

[https://studio.azureml.net](https://studio.azureml.net/)



Biểu diễn dữ liệu bằng đồ họa

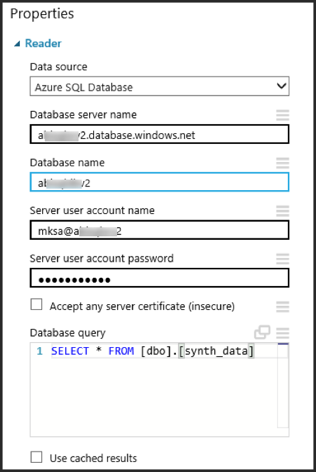




Truy vấn dữ liệu

Thực hiện câu lệnh

SELECT \* FROM [dbo].[synth\_data]



**Bài thực hành số 9: Thực thi công cụ R, nhúng Python trên AzureML**

**Nội dung:**

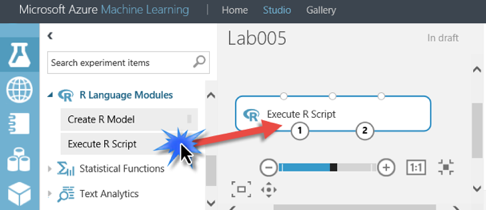
* Nhúng R vào AzureML
* Nhúng Python vào AzureML

**Yêu cầu:**

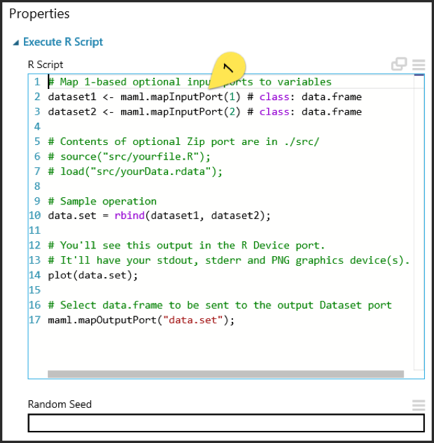
Đã cài đặt các công cụ ở buổi 6

**Chi tiết:**

Chạy Rstudio, tạo 1 blank Azure ML experiment



Gõ script



Bổ sung:

# Generate synthetic data

x <- seq(1, 30)

y <- x

noise <- runif(30, -1, 1)

ywnoise <- y + noise \* 2

# plot point cloud on a chart

plot(x, ywnoise, xlab = NA, ylab = NA)

# combine two columns to create data grid

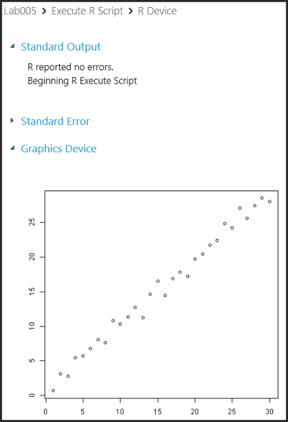
linoise <- cbind(x, ywnoise)

linoise <- as.data.frame(linoise)

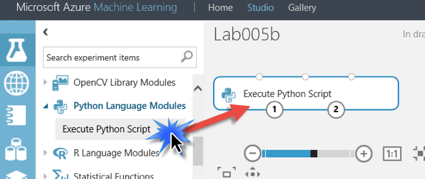
# Select data.frame to be sent to the output Dataset port

maml.mapOutputPort("linoise");

Kiểm tra kết quả



Nhúng Python



Gõ scripts

import matplotlib

matplotlib.use('agg')

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

def azureml\_main(dataframe1=None, dataframe2=None):

x = range(1, 31)

y = x

noise = np.random.uniform(-1, 1, 30)

ywnoise = y + noise \* 2

d = {'x' : np.asarray(x), 'ywnoise' : ywnoise}

linoise = pd.DataFrame(d)

fig = plt.figure()

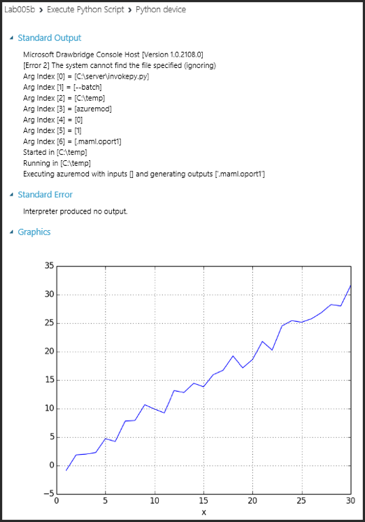
ax = fig.gca()

linoise.plot(kind='line', ax=ax, x='x', y='ywnoise')

fig.savefig('linoise.png')

return linoise

Kiểm chứng kết quả



**Bài thực hành số 10. Kiểm tra đồ án**