

Regression in Practice: A Python Practicum Lecture 6b: 2023-02-13

MAT A35 – Winter 2023 – UTSC

Prof. Yun William Yu

Try it out: Python basics

```
x = 123
```

```
y = 234
```

```
z = 345
```

```
print(x / y * z - x)
```

```
print(x / y * z - x^2)
```

A: 42

B: 50.29619392032321

C: 58.34615384615384

D: TypeError

E: None of the above

Try it out: numpy

- `import numpy as np`
- `X = np.array([1,2,3,4,5])`
- `Y = X`
- `print(X*Y)`

A: 54

B: [1 4 9 16 25]

C: $\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 6 & 8 & 10 \\ 3 & 6 & 9 & 12 & 15 \\ 4 & 8 & 12 & 16 & 20 \\ 5 & 10 & 15 & 20 & 25 \end{bmatrix}$

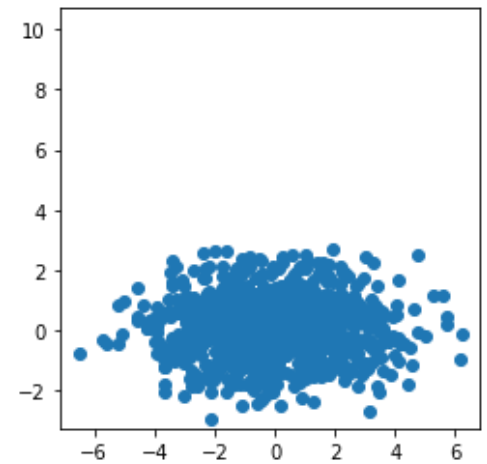
D: TypeError

E: None of the above

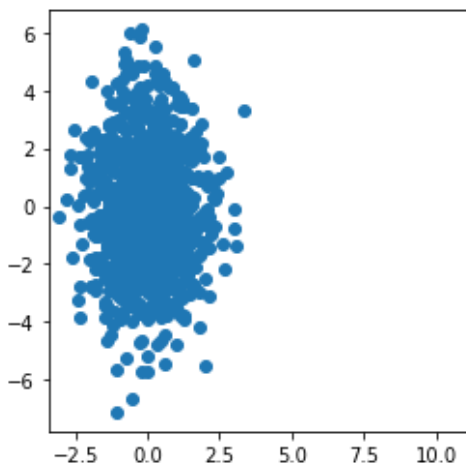
Try it out: matplotlib

- `Xrand = np.random.randn(1000)`
- `Yrand = np.random.randn(1000)`
- `plt.scatter(Xrand, Yrand)`
- `plt.axis('square')`
- `plt.show()`

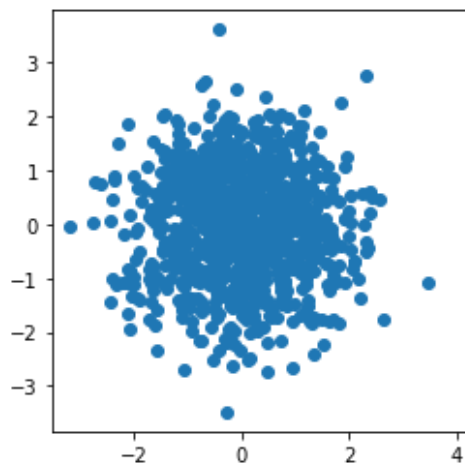
A



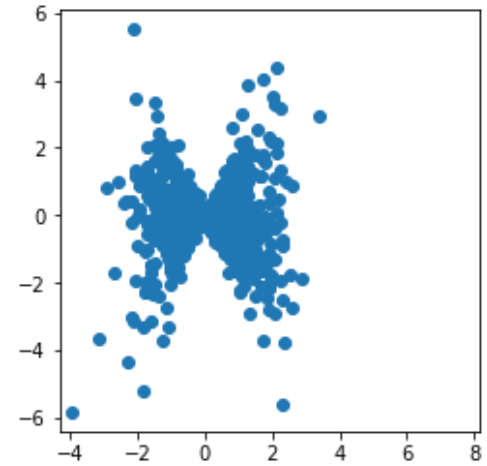
B



C



D



Try it out: fitting a model

- Suppose Alice is taking MATA35 and has taken the first four quizzes, with marks of 52, 63, 68, and 74 on quizzes 1, 2, 3, and 4 respectively.
- Alice decides to use a linear regression model to predict her score on quiz 5. What does she predict to be her mark?

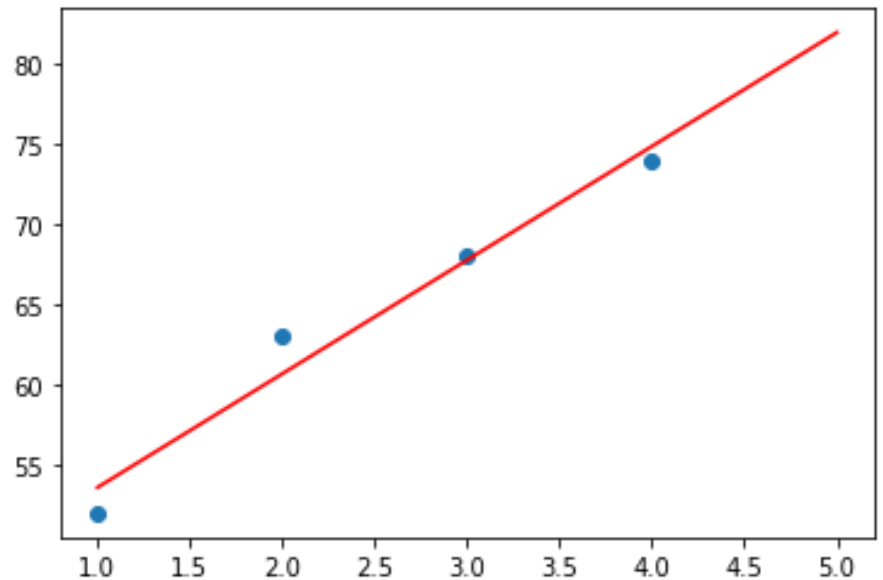
A: 80

B: 81

C: 82

D: 83

E: None of the above



Polynomial regression

- What if our data doesn't look linear?

Different types of regression

- Linear regression:

$$f(x) = mx + b$$

- Quadratic regression:

$$f(x) = m_2x^2 + m_1x + b$$

- Cubic regression:

$$f(x) = m_3x^3 + m_2x^2 + m_1x + b$$

- Polynomial regression of degree n :

$$f(x) = b + \sum_{i=1}^n m_i x^i$$

- Exponential regression:

$$f(x) = b + m_0 e^{m_1 x}$$

Be careful about too many parameters

- The more parameters you have (e.g. in a polynomial regression), the better your mean squared error will be.
- However, sometimes, you will overfit to the data.