

# simpleregression

November 4, 2024

```
[23]: import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

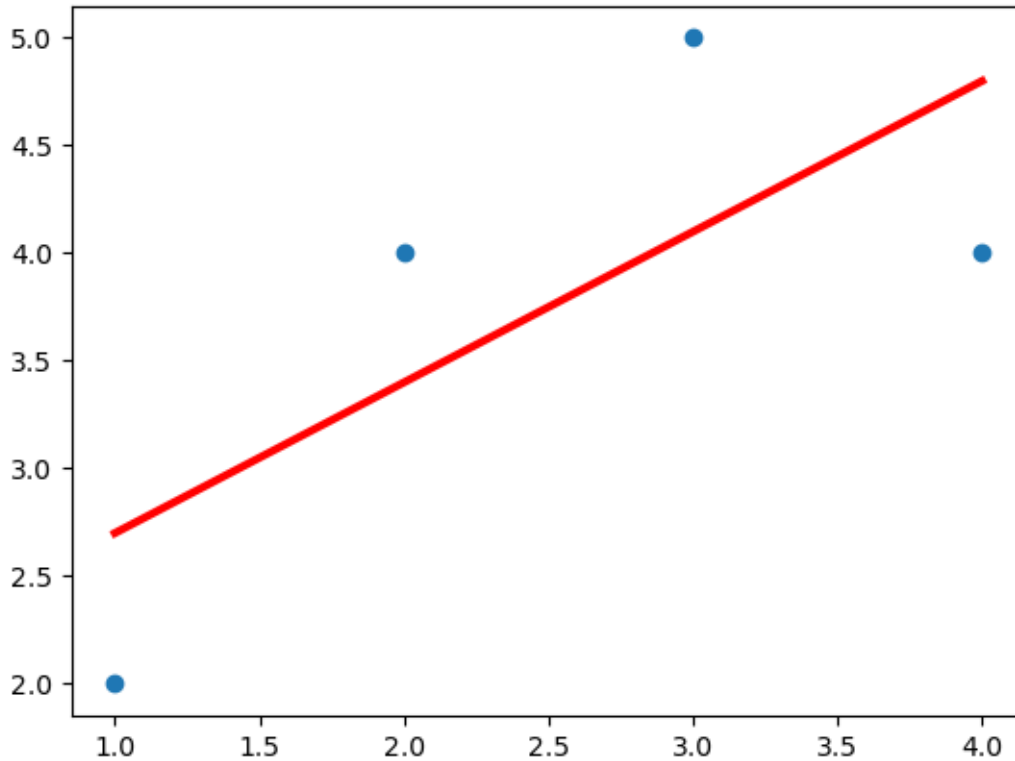
# Créez vos données d'entraînement
# Supposons que vous ayez une variable indépendante X et une variable dépendante
→y
X = np.array([[1], [2], [3], [4]])
y = np.array([2, 4, 5, 4])

# Créez le modèle de régression linéaire
modele_regression = LinearRegression()

# Entraînez le modèle avec vos données
modele_regression.fit(X, y)

# Une fois le modèle entraîné, vous pouvez l'utiliser pour faire des prédictions
# Par exemple, pour prédire une valeur pour X=5
prediction = modele_regression.predict([[5]])
plt.scatter(X, y)
plt.plot(X,modele_regression.predict(X), c='red', lw =3)
```

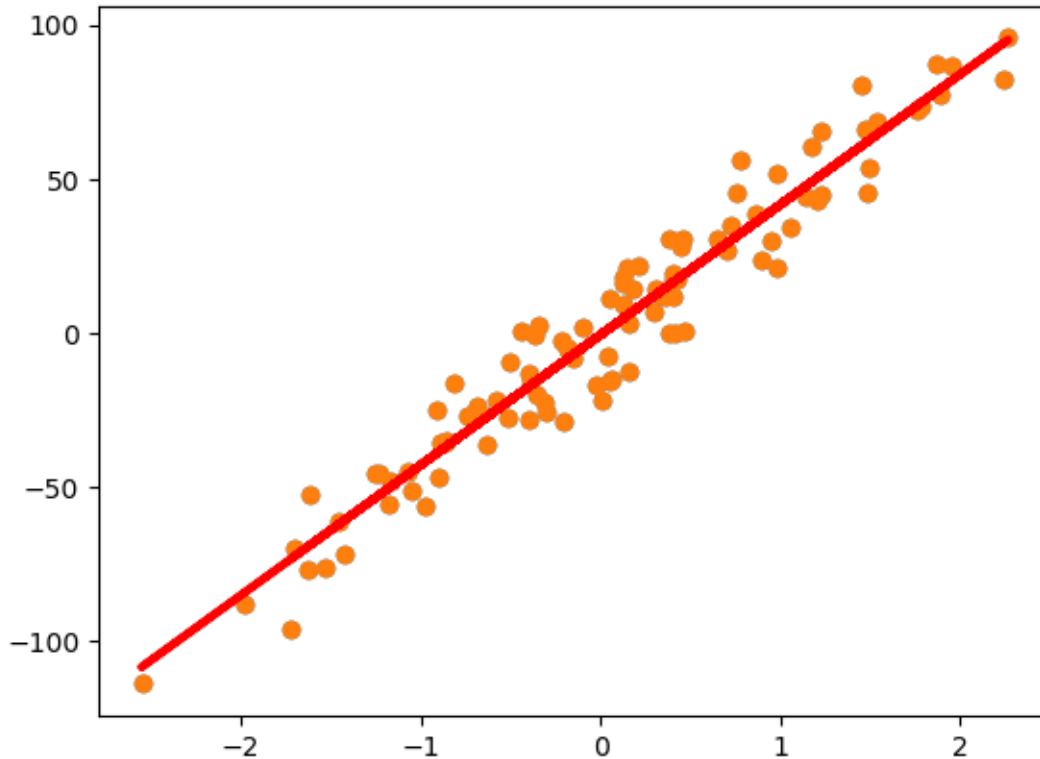
```
[23]: [<matplotlib.lines.Line2D at 0x21763c133e0>]
```



```
[19]: import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_regression
from sklearn.linear_model import SGDRegressor
np.random.seed(0)
x, y = make_regression(n_samples=100, n_features=1, noise=10)
plt.scatter(x, y)
model = SGDRegressor(max_iter=1000, eta0=0.001)
model.fit(x,y)
print('Coeff R2 =', model.score(x, y))
plt.scatter(x, y)
plt.plot(x,model.predict(x), c='red', lw =3)
```

Coeff R2 = 0.9416560747789743

```
[19]: [<matplotlib.lines.Line2D at 0x21768d33f20>]
```



```
[21]: from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
# initialisation du modèle
regression_model = LinearRegression()
regression_model.fit(x, y)
# Prédiction
y_predicted = regression_model.predict(x)
# Évaluation du modèle
rmse = mean_squared_error(y, y_predicted)
r2 = r2_score(y, y_predicted)
# Affichage des valeurs
print("Pente : ", regression_model.coef_)
print("Ordonnée à l'origine : ", regression_model.intercept_)
print("Racine carrée de l'erreur quadratique moyenne : ", rmse)
print('Score R2 : ', r2)
# Tracée des valeurs
# Points de données
plt.scatter(x, y, s=10)
plt.xlabel('x')
plt.ylabel('y')
# Valeurs prédites
plt.plot(x, y_predicted, color='r')
```

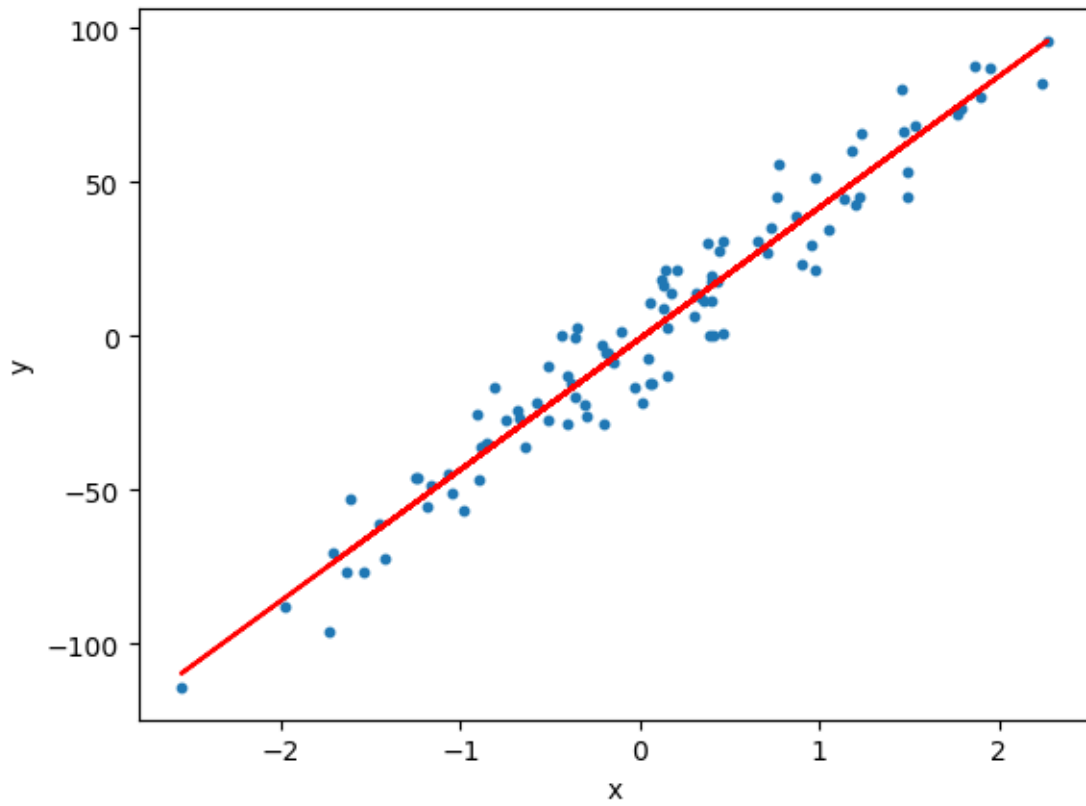
```
plt.show()
```

Pente : [42.61943029]

Ordonnée à l'origine : -0.8141818270307257

Racine carrée de l'erreur quadratique moyenne : 114.17148616819482

Score R2 : 0.9417294727711081



```
[ ]:
```