This is the solution for the homework assignment of the Machine Learning and Optimization lecture for WS2023.

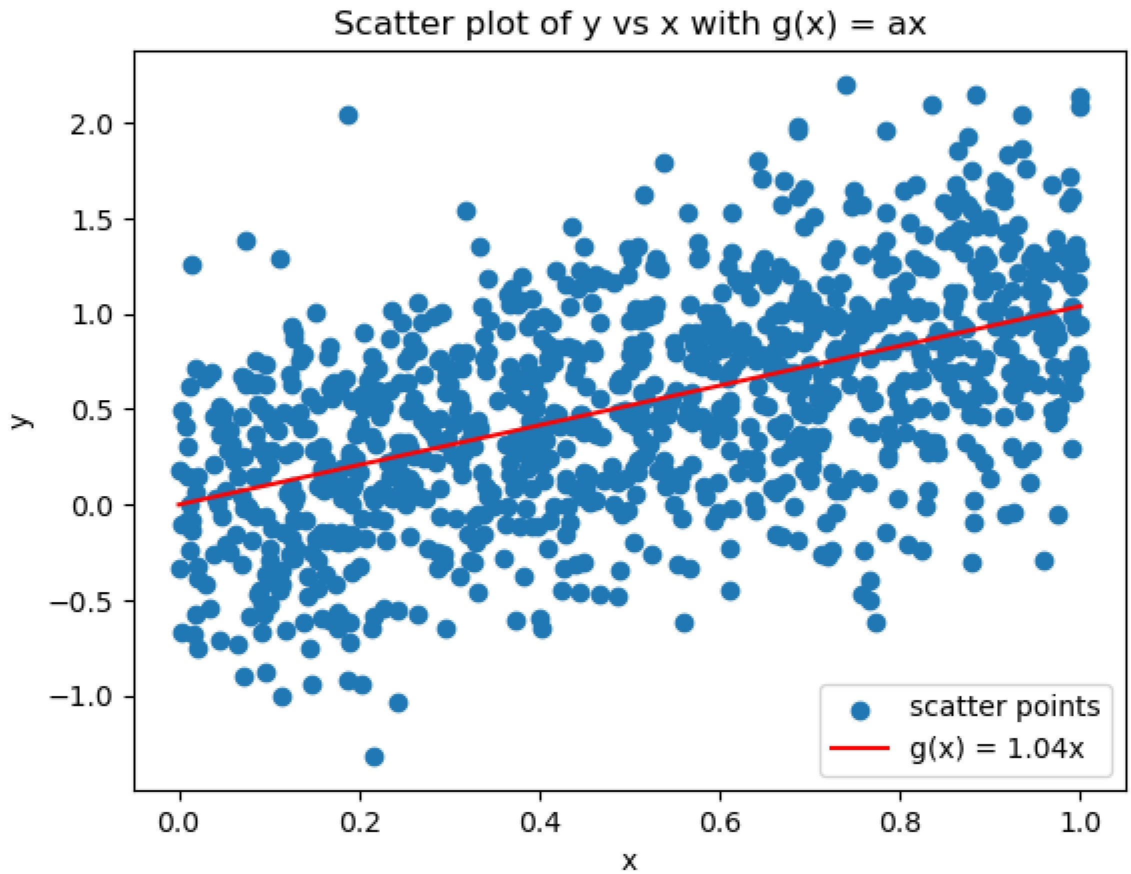
In [1]:
```
import numpy as np
from matplotlib import pyplot as plt
```

Problem 1. and 2.

In [2]:
```
# n = 1000 points, x_i uniformly distributed on the interval [0,1].
n = 1000
x = np.random.uniform(0, 1, n)

# compute y_i and epsilon
e = np.random.normal(0, np.sqrt(0.25), n)
y = x + e
```

In [3]:
```
# compute the minimum of f(a)
# make df(a)/da = 0 Then,  a = \sum x_i*y_i / \sum x_i^2
a = np.sum(x * y) / np.sum(x**2)
```

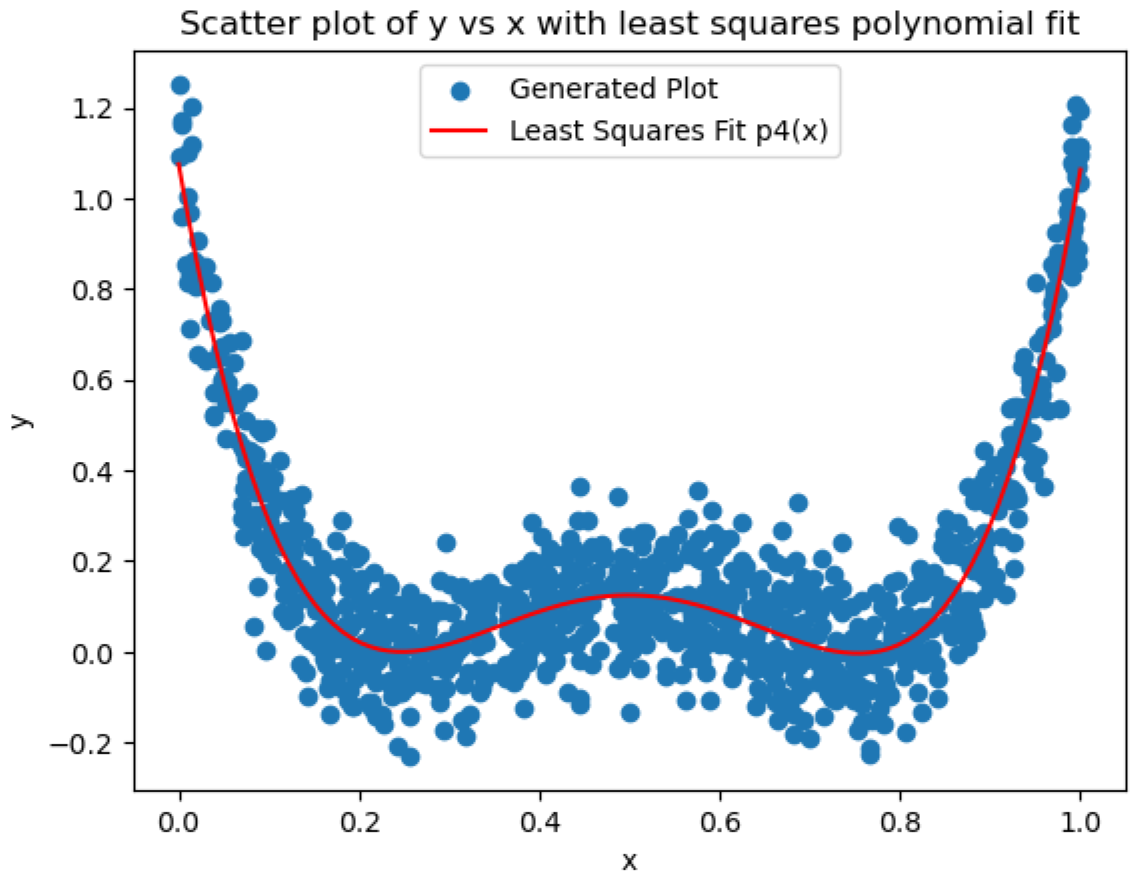
In [4]:
```
plt.scatter(x, y,  label='scatter points')

# add line g(x) = ax
plt.plot([0, 1], [0, a], '-r', label=f'g(x) = {a:.2f}x')

plt.legend()
plt.xlabel('x')
plt.ylabel('y')
plt.title('Scatter plot of y vs x with g(x) = ax')
plt.show()
```



Problem 4.

In [5]:
```
# Generate additionally variables
xi = np.random.normal(0, np.sqrt(0.01), n)
y = 30*(x - 0.25)**2*(x - 0.75)**2 + xi

# build polynomial p4(x), cofficient a
X = np.stack([x**i for i in range(5)], axis=1)
a = np.linalg.inv(X.T @ X) @ X.T @ y

# polynomial value fitting
# we need a0, a1x, a2x^2, a3x^3, a4x^4, so range(5)
x_fit = np.linspace(0, 1, 100)
y_fit = sum(a[i] * x_fit**i for i in range(5))
```

In [6]:
```
plt.scatter(x, y, label='Generated Plot')
plt.plot(x_fit, y_fit, '-r', label='Least Squares Fit p4(x)')
plt.legend()
plt.xlabel('x')
plt.ylabel('y')
plt.title('Scatter plot of y vs x with least squares polynomial fit')
plt.show()
```



Scatter plot of y vs x with least squares polynomial fit
Generated Plot
Least Squares Fit p4(x)
y
x
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0