Problem 4 - via Julia

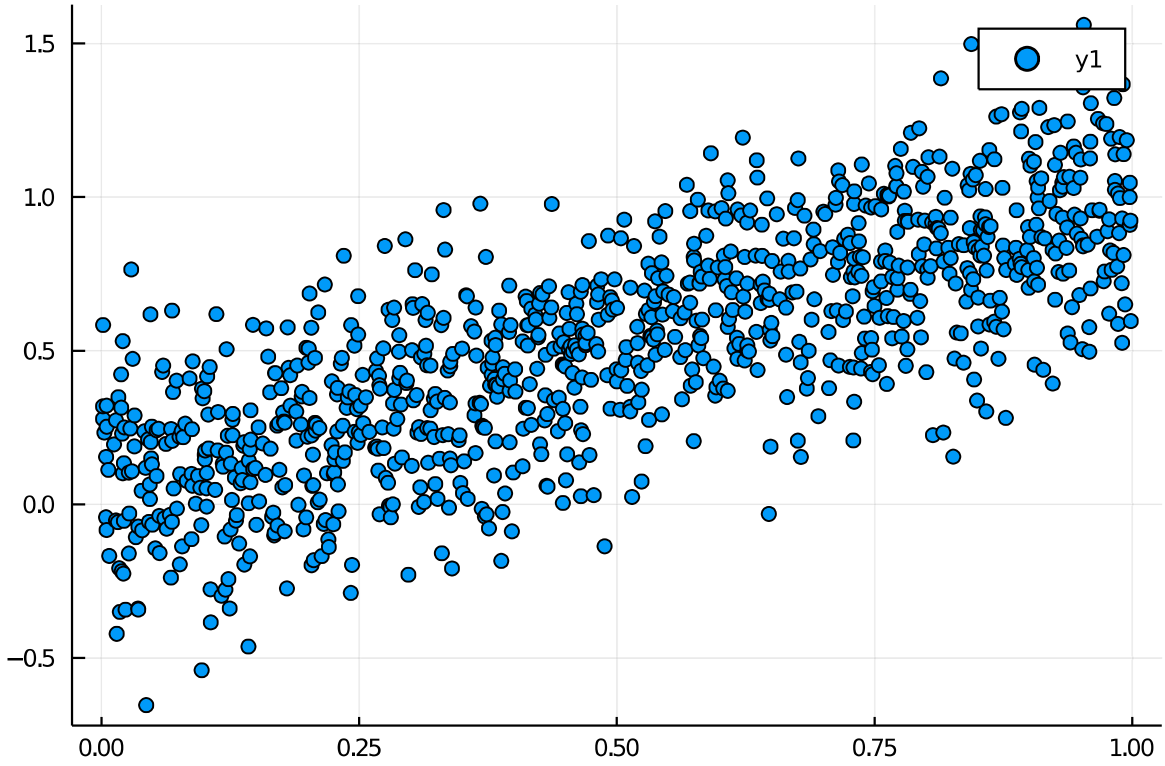
In [1]:

```julia
using LinearAlgebra, Plots
```

In [2]:

```julia
# 1)
n = 1000
x = rand(n)                                 # uniform over [0,1]
ϵ = 0.25 .* randn(n)                        # standardnormal * 0.25
y = x + ϵ
p = sortperm(x)                             # sort series data
x, y = x[p], y[p]
fig = scatter(x, y)
```

Out[2]:

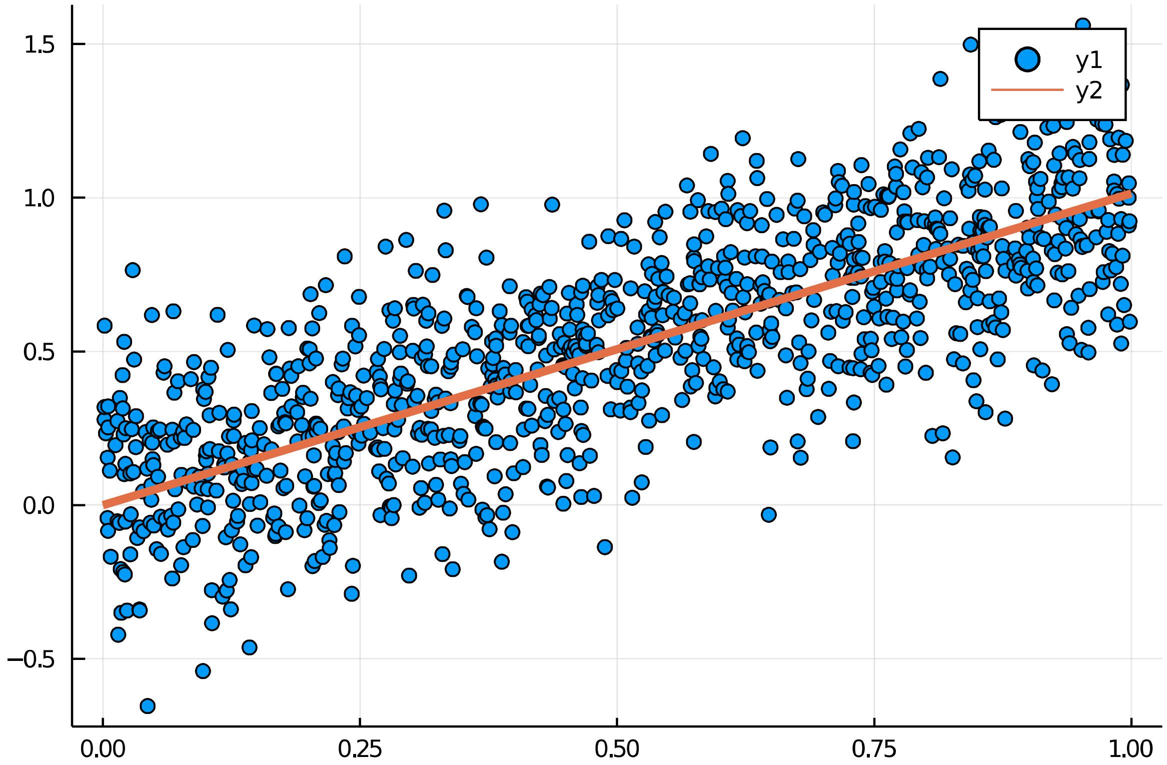


In [3]:

```julia
# 2)
f(a) = sum(@. (x * a - y)^2)
Df(a) = sum(@. 2 * x * (x * a - y))
DDf = sum(@. 2 * x^2)
a, h = 0.5, 0
f_new, f_old = f(a), Inf
while f_old - f_new > eps()                 # simple newton-method
    h = -Df(a)/DDf
    @show a = a + h
    f_old = f_new
    @show f_new = f(a)
end
fig = plot!(0:0.5:1, x -> a*x, linewidth=4)
```

```
a = a + h = 1.014284724543708
f_new = f(a) = 56.27862653553691
a = a + h = 1.0142847245437083
f_new = f(a) = 56.27862653553691
```

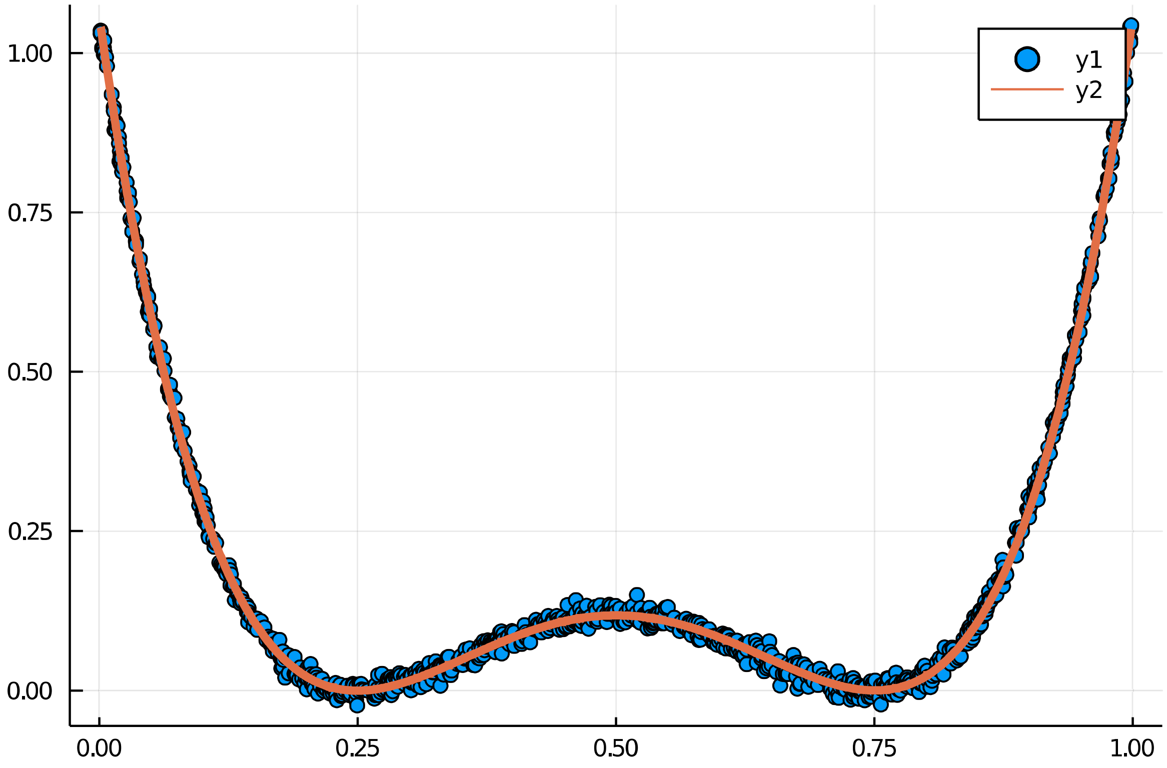
Out[3]:



In [5]:

```
# 4)
d = 4
ϵ = 0.01 .* randn(n)
y = @. 30 * (x - .25)^2 * (x - .75)^2 + ϵ
fig2 = scatter(x, y)
X = hcat([x.^m for m in 0:d]...)            # flatten arrays
a = (X'X) \ (X'y)
fig2 = plot!(x, z -> sum(a[i+1] * z^(i) for i in 0:d), linewidth=4)
```

Out[5]:



1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
● y1
y2