Table of Contents

In [1]:
```
using LinearAlgebra, Random, Plots
```

In [2]:
```
"""
    A = sbm(n, p, q)

The adjacency matrix of a random graph with 2n vertices,
divided into groups 1:n and n+(1:n).
The probability of edges inside groups is p, and between groups is q.
"""
function sbm(n::Int, p, q)
    J = ones(n,n)
    A = triu([p*J q*J; q*J p*J],1)
    M = Float64.(rand(2n,2n) .< A)
    M = M + M'
end
```

Out[2]:
```
sbm
```

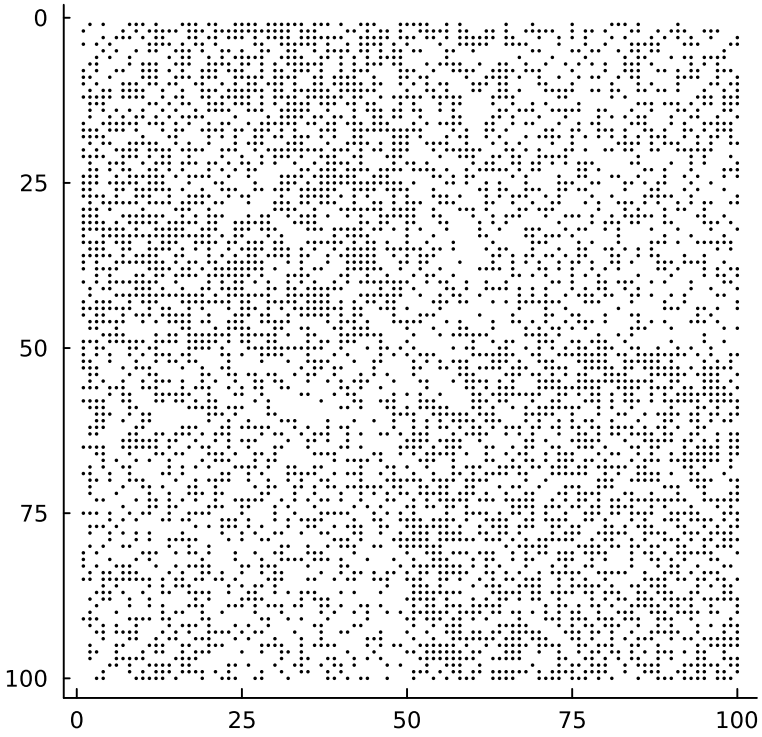
In [3]:
```
Random.seed!(1)
sbm(3,0.99,0.01)
```

Out[3]:
```
6×6 Matrix{Float64}:
 0.0  1.0  1.0  0.0  0.0  0.0
 1.0  0.0  1.0  0.0  0.0  0.0
 1.0  1.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  1.0  1.0
 0.0  0.0  0.0  1.0  0.0  1.0
 0.0  0.0  0.0  1.0  1.0  0.0
```

If p and q are far enough apart, you can see the clusters when we make them consecutive. In the following plot we show the adjacency matrix, with black dots representing edges.

In [4]:
```
n = 50
p = 0.6
q = 0.4
M = sbm(n, p, q)
spy(M)
```

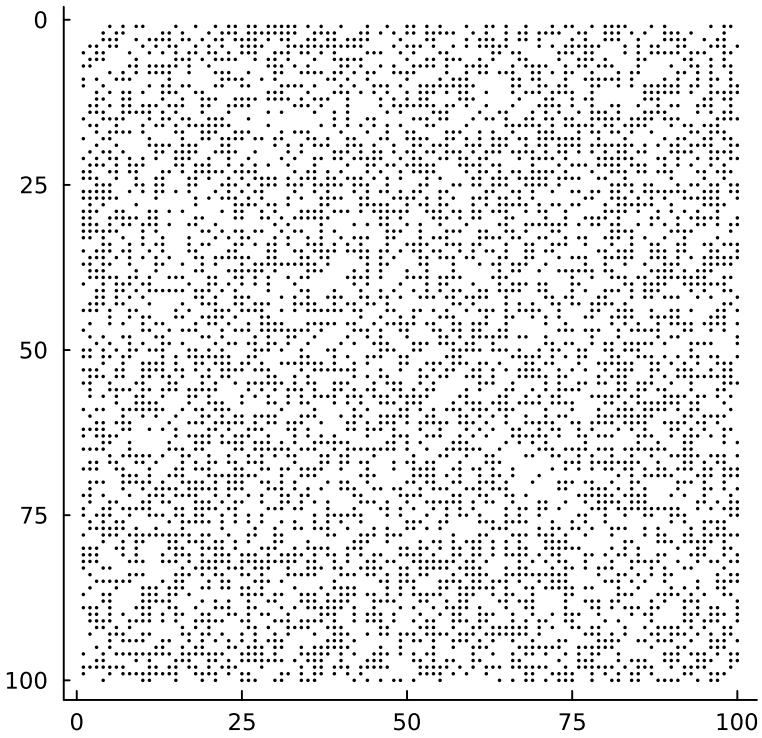
Out[4]:



But, you can't see them if they are scrambled.

In [5]:
```
perm = randperm(2n)
spy(M[perm,perm])
```

Out[5]:

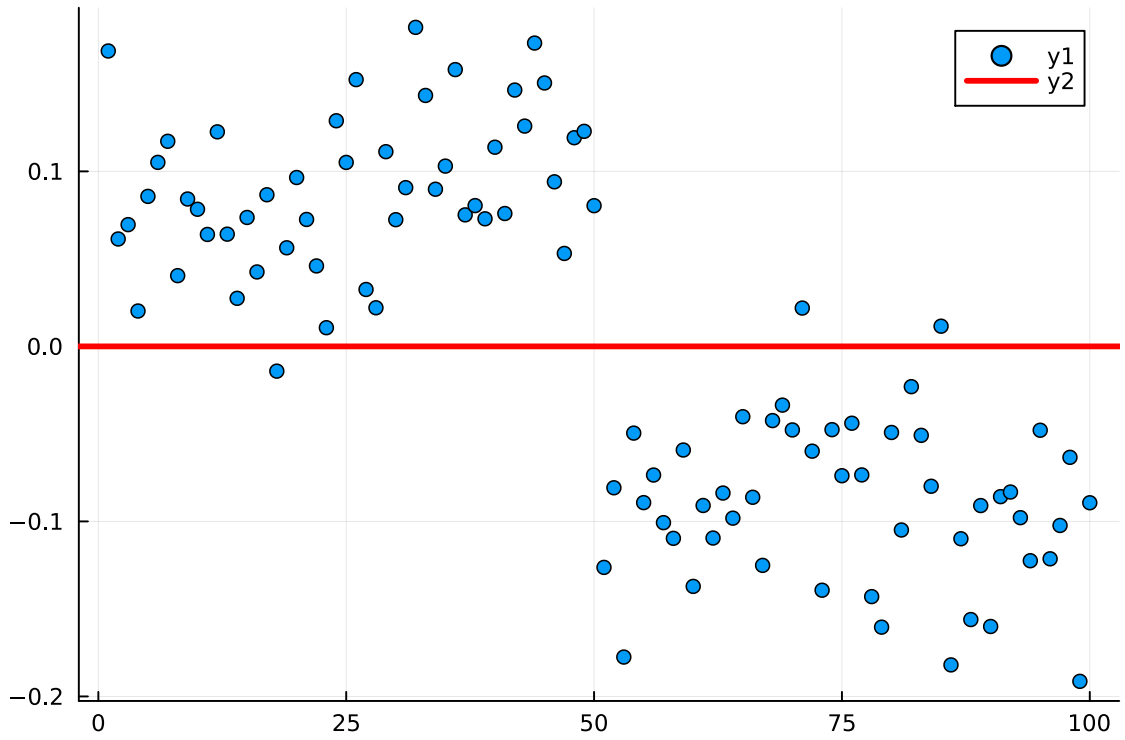


In [6]:
```
E = eigvals(M)
```

Out[6]:
```
100-element Vector{Float64}:
 -10.29791654041372
  -9.472167647138816
  -9.091274494302471
  -8.940088672192948
  -8.731812513359465
  -8.610189012183405
  -8.124274547790662
  -7.635371578858221
  -7.557814253581181
  -7.390984372090813
  -7.216420774650352
  -6.970648187872823
  -6.840254960008986
   ⋮
   5.9853944765849185
   6.106475392190413
   6.39940939442177
   6.596900713467748
   6.841122953591303
   7.025768146347776
   7.356279554461459
   7.760434130214361
   8.21147127214954
   8.892623092168565
  11.32882738404859
  49.9412805397874
```

In [7]:
```
psi2 = eigvecs(M)[:,end-1]
psi2 = sign(sum(psi2[1:n]))*psi2
pl = scatter(psi2)
hline!(pl, [0], color=:red, linewidth=3)
```

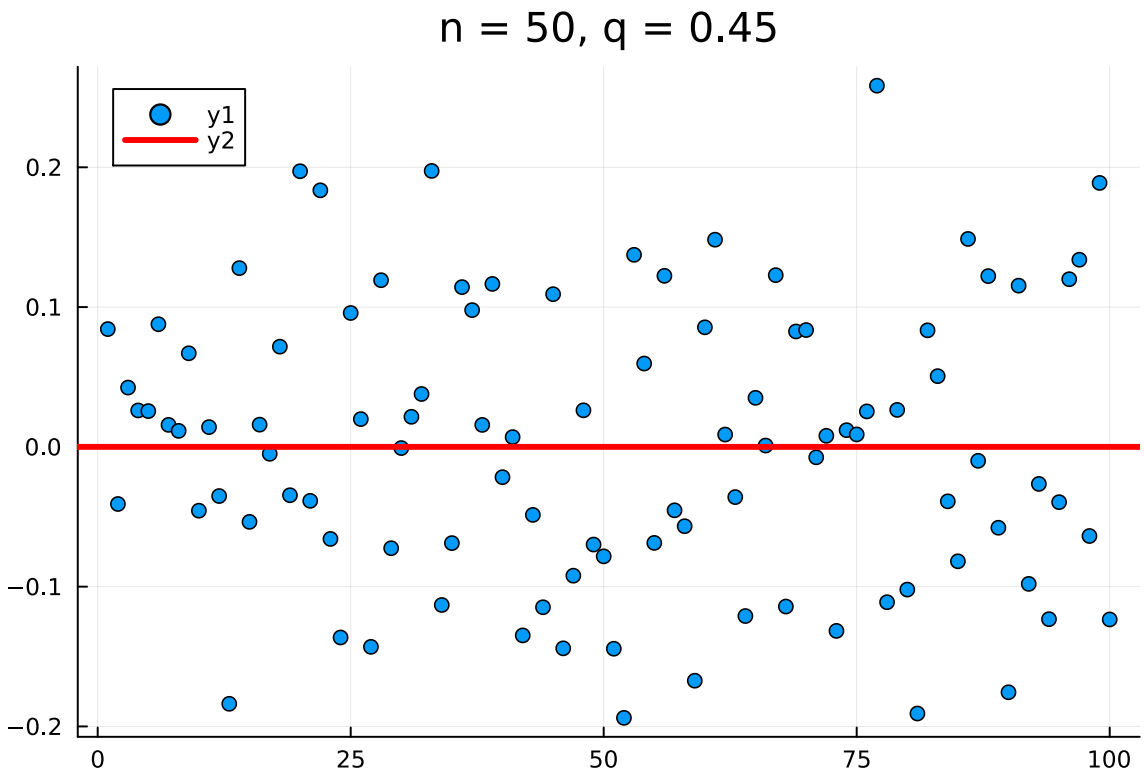
Out[7]:



In [8]:
```
# compute the number misclassified in each group
sum(psi2[1:n] .< 0), sum(psi2[(n+1):2n] .> 0)
```

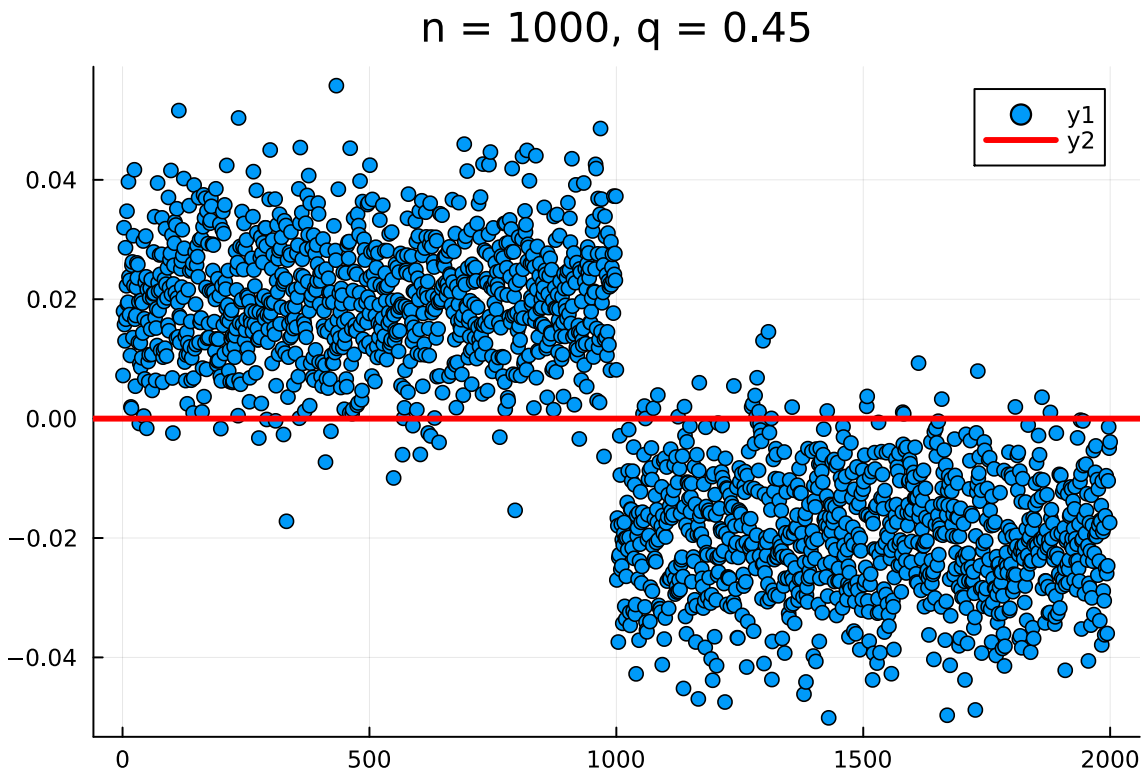
Out[8]: (1, 2)

In [9]:
```
p = 0.5
q = 0.45
M = sbm(n, p, q)
psi2 = eigvecs(M)[:,end-1]
psi2 = sign(sum(psi2[1:n]))*psi2
pl = scatter(psi2, title="n = $n, q = $q")
hline!(pl, [0], color=:red, linewidth=3)
display(pl)
# compute the number misclassified in each group
sum(psi2[1:n] .< 0), sum(psi2[(n+1):2n] .> 0)
```



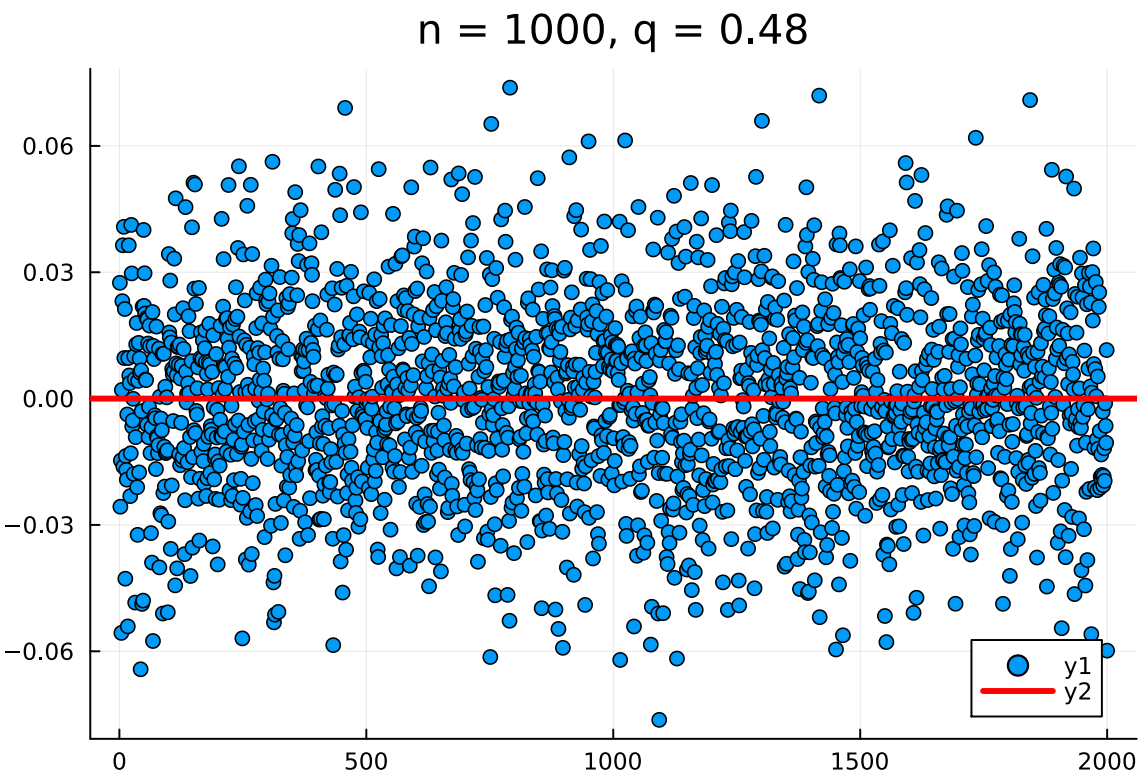
Out[9]: (23, 25)

In [10]:
```
p = 0.5
q = 0.45
n = 1000
M = sbm(n, p, q)
psi2 = eigvecs(M)[:,end-1]
psi2 = sign(sum(psi2[1:n]))*psi2
pl = scatter(psi2, title="n = $n, q = $q")
hline!(pl, [0], color=:red, linewidth=3)
display(pl)

# compute the number misclassified in each group
sum(psi2[1:n] .< 0), sum(psi2[(n+1):2n] .> 0)
```



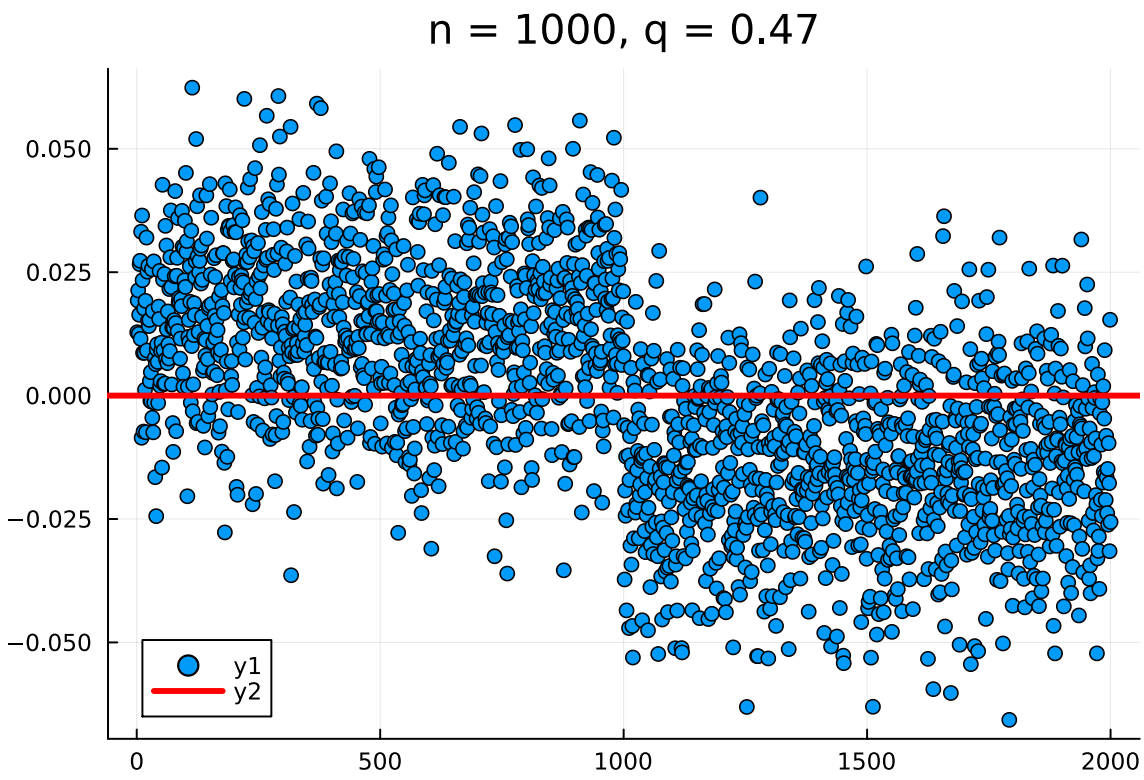
Out[10]: (22, 30)

In [11]:
```
p = 0.5
q = 0.48
n = 1000
M = sbm(n, p, q)
psi2 = eigvecs(M)[:,end-1]
psi2 = sign(sum(psi2[1:n]))*psi2
pl = scatter(psi2, title="n = $n, q = $q")
hline!(pl, [0], color=:red, linewidth=3)
display(pl)

# compute the number misclassified in each group
sum(psi2[1:n] .< 0), sum(psi2[(n+1):2n] .> 0)
```



Out[11]: (480, 478)

In [12]:
```
p = 0.5
q = 0.47
n = 1000
M = sbm(n, p, q)
psi2 = eigvecs(M)[:,end-1]
psi2 = sign(sum(psi2[1:n]))*psi2
pl = scatter(psi2, title="n = $n, q = $q")
hline!(pl, [0], color=:red, linewidth=3)
display(pl)

# compute the number misclassified in each group
sum(psi2[1:n] .< 0), sum(psi2[(n+1):2n] .> 0)
```

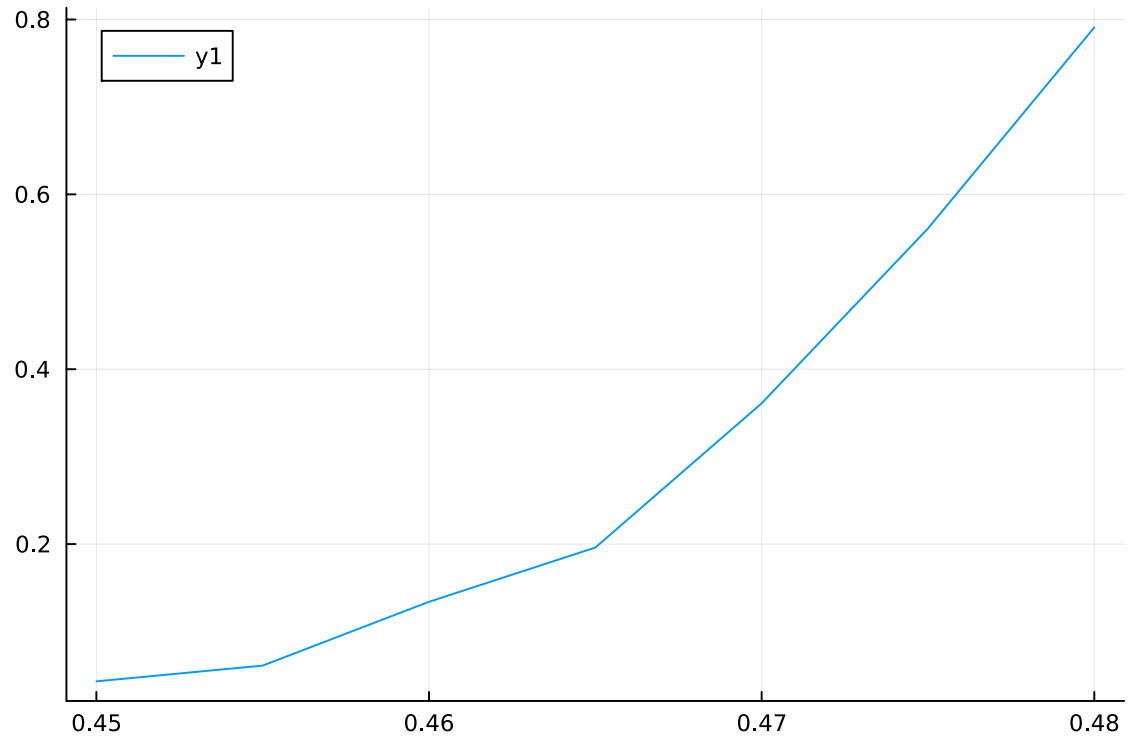


Out[12]: (193, 189)

Plot number misclassified as a function of q.

In [13]:
```
Random.seed!(0)
qs = 0.45:0.005:0.48
misclass = []
for q in qs

    M = sbm(n, p, q)
    E = eigen(M)
    psi2 = E.vectors[:,end-1]
    psi2 = sign(sum(psi2[1:n]))*psi2

    push!(misclass, sum(psi2[1:n] .< 0) + sum(psi2[(n+1):2n] .> 0))
end

plot(qs, misclass ./ n)
```

Out[13]:

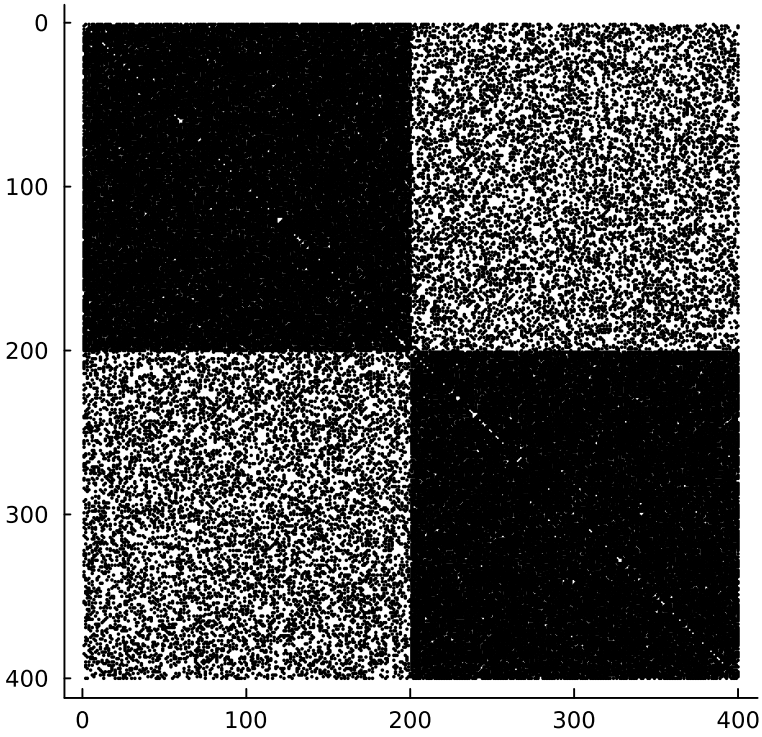


In [14]:
```
[qs (misclass ./ n)]
```

Out[14]:
```
7×2 Matrix{Float64}:
 0.45   0.043
 0.455  0.061
 0.46   0.134
 0.465  0.196
 0.47   0.361
 0.475  0.561
 0.48   0.791
```

In [15]:
```
using Laplacians
n = 200
p = 0.8
q = 0.2
M = sbm(n, p, q)
spy(M)
```

Out[15]:

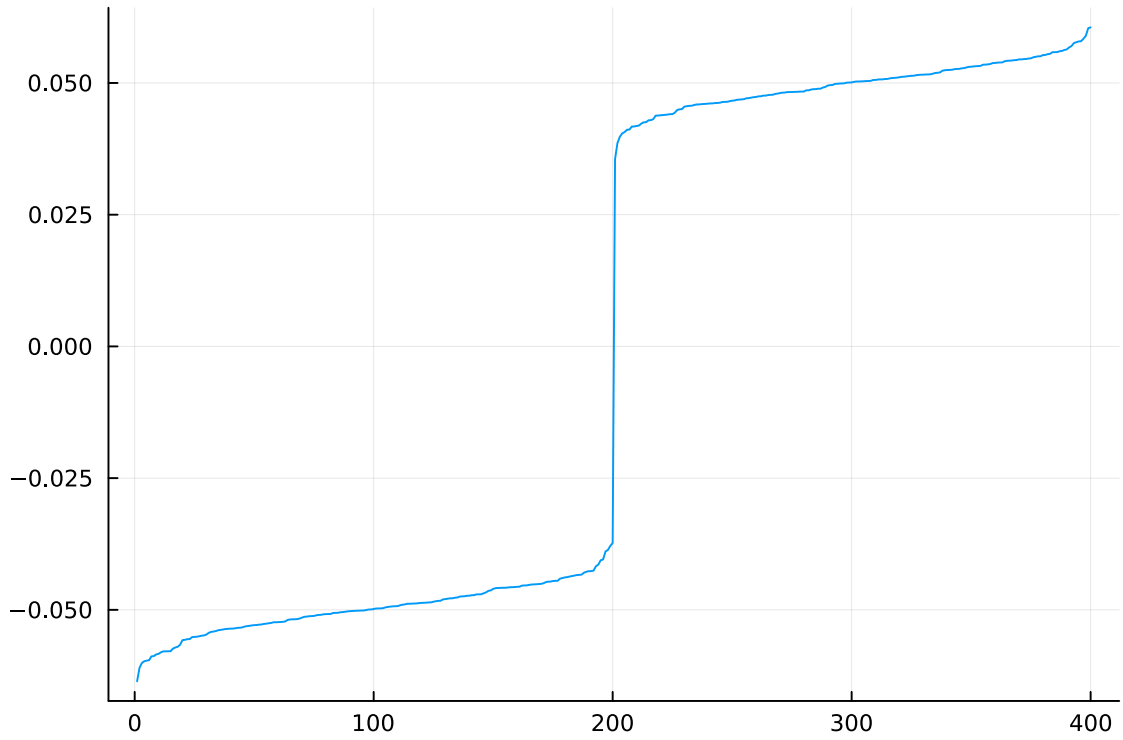


In [16]:
```
using SparseArrays
L = lap(M)
val, v = fiedler(sparse(M))
```

Out[16]:
```
([79.39474129281888], [0.04391062499540743; 0.04114472869265903; … ; -0.051
2779007238633; -0.04562299675077166;;], 1, 1, 20, [3.4956241482370324e-6,
1.586644265732379e-5, 7.035397482345095e-6, 6.093943254004973e-6, 3.8311290
536797555e-5, -2.9267631805712988e-5, -7.678628007454413e-6, -1.40982874300
98228e-5, 2.8782217922299035e-5, 2.8279473723023937e-6  …  6.65559834588915
7e-6, 3.288421883547046e-6, 3.1428266978666745e-5, 5.746031997540853e-7, 1.
0989982857081415e-5, -3.401987509614625e-6, -1.8983417910803468e-5, 1.47613
4643900857e-5, -2.2531843221225923e-6, 8.935822085352649e-6])
```

In [17]:
```
plot(sort(vec(v)),label=false)
```

Out[17]:

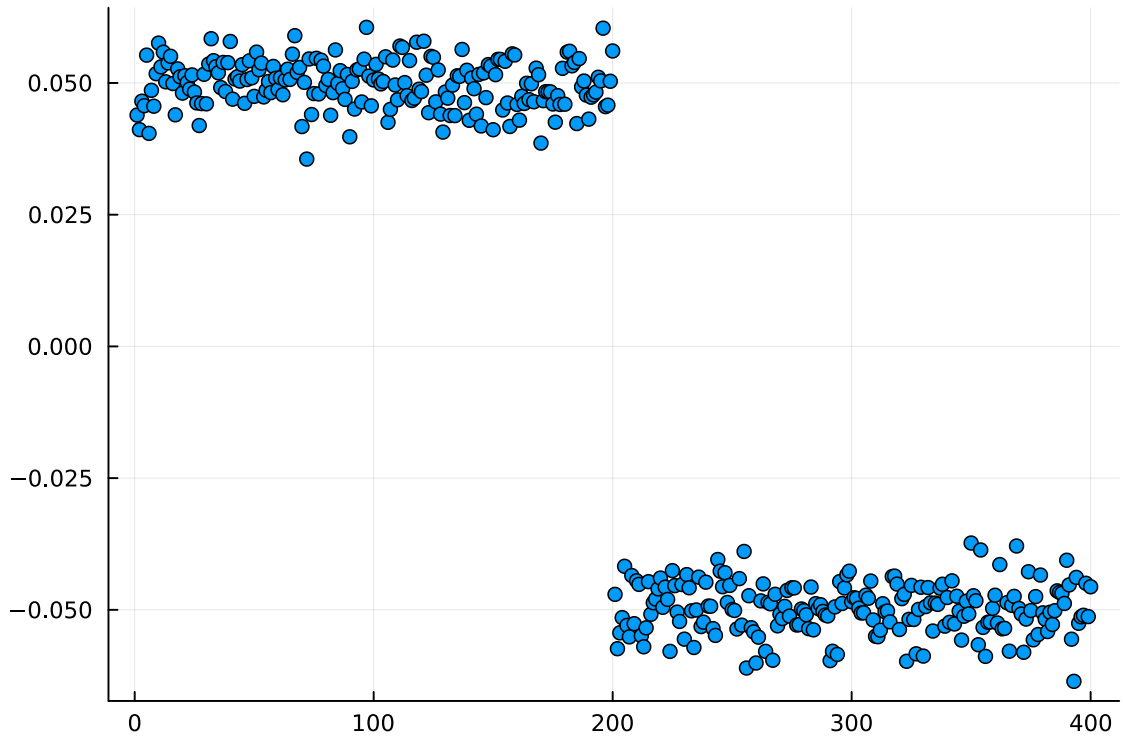


In [18]:
```
savefig("sbm_fiedler.pdf")
```

Out[18]:
```
"/Users/spielman/Library/CloudStorage/Dropbox/gits/462/julia/sbm_fiedler.pdf"
```

In [19]:
```
scatter(vec(v),label=false)
```

Out[19]:



In [20]:
```
savefig("sbm_fiedler_scat.pdf")
```

Out[20]:
```
"/Users/spielman/Library/CloudStorage/Dropbox/gits/462/julia/sbm_fiedler_scat.pdf"
```

In [ ]:
```

```