Ctrl + S

```
PlotlyBackend()
```

```
using Plots;plotly()
```

```
For saving to png with the `Plotly` backend `PlotlyBase` and `PlotlyKaleido` need to be installed.
err:
    ArgumentError("Package PlotlyBase not found in cur" … 52 bytes … " to instal
```

```
using DelimitedFiles
```

```
whitewine = "D:/University/DT Coding/ML Julia/Homework/A4/winequality-white.csv"
```

```
whitewine = "D:/University/DT Coding/ML Julia/Homework/A4/winequality-white.csv"
```

```
A = 4899×12 Matrix{Any}:
 "fixed acidity"   "volatile acidity"  …   "sulphates"    "alcohol"   "quality"
 7                 0.27                    0.45           8.8         6
 6.3               0.3                     0.49           9.5         6
 8.1               0.28                    0.44          10.1         6
 7.2               0.23                    0.4            9.9         6
 7.2               0.23                …   0.4            9.9         6
 8.1               0.28                    0.44          10.1         6
 ⋮                                     ⋱                 ⋮
 6.5               0.23                    0.54           9.7         5
 6.2               0.21                    0.5           11.2         6
 6.6               0.32                …   0.46           9.6         5
 6.5               0.24                    0.46           9.4         6
 5.5               0.29                    0.38          12.8         7
 6                 0.21                    0.32          11.8         6
```

```
A = readdlm(whitewine, ';')
```

```
redwine = "D:/University/DT Coding/ML Julia/Homework/A4/winequality-red.csv"
```

```
redwine = "D:/University/DT Coding/ML Julia/Homework/A4/winequality-red.csv"
```

```
B = 1600×12 Matrix{Any}:
  "fixed acidity"   "volatile acidity"  …   "sulphates"    "alcohol"   "quality"
  7.4               0.7                     0.56           9.4         5
  7.8               0.88                    0.68           9.8         5
  7.8               0.76                    0.65           9.8         5
 11.2               0.28                    0.58           9.8         6
  7.4               0.7                 …   0.56           9.4         5
  7.4               0.66                    0.56           9.4         5
  ⋮                                     ⋱                 ⋮
  6.8               0.62                    0.82           9.5         6
  6.2               0.6                 …   0.58          10.5         5
  5.9               0.55                    0.76          11.2         6
  6.3               0.51                    0.75          11           6
  5.9               0.645                   0.71          10.2         5
  6                 0.31                    0.66          11           6
```

```
B = readdlm(redwine, ';')
```

```
readData (generic function with 1 method)
```

```
function readData(path)
    A = readdlm(path, ';')
    y = float.(A[2:end,end])
    X = float.([ones(length(y)) A[2:end, 1:end-1]])
    return X, y
end
```

```
(4898×12 Matrix{Float64}:                                                , [6.0, 6.0, 6
  1.0  7.0  0.27  0.36  20.7  0.045  45.0  170.0  1.001    3.0   0.45   8.8
```

```
X1, y1 = readData(whitewine)
```

```
(1599×12 Matrix{Float64}:                                                , [5.0, 5.0, 5
  1.0   7.4  0.7    0.0   1.9  0.076  11.0  34.0  0.9978   3.51  0.56   9.4
```

```
X2, y2 = readData(redwine)
```

```
train (generic function with 1 method)
```

```
function train(X,y)
    return inv(X'*X)*X'*y
end
```

```
θ_white =
  [150.193, 0.06552, -1.86318, 0.0220902, 0.0814828, -0.247277, 0.00373277, -0.000285747, -1
```

```
θ_white = train(X1, y1)
```

```
θ_red =
  [21.9652, 0.0249906, -1.08359, -0.182564, 0.0163313, -1.87423, 0.00436133, -0.00326458, -1
```

```
θ_red = train(X2, y2)
```

```
predict (generic function with 1 method)
```

```
predict(θ, xNew) = xNew*θ
```

```
ŷ_red =
  [4.16865, 4.04459, 4.20466, 5.04748, 4.16865, 4.24102, 4.37724, 4.67738, 4.49787, 5.15678,
```

```
ŷ_red = predict(θ_white, X2)
```

```
ŷ_white =
  [5.37973, 5.23388, 5.67508, 5.44423, 5.44423, 5.67508, 5.43781, 5.37973, 5.23388, 5.84202,
```

```
ŷ_white = predict(θ_red, X1)
```

```
classify (generic function with 2 methods)
```

```
function classify(ŷ, y, ϵ = 0.5)
    error = abs.(ŷ - y)
    return error .< ϵ
end
```

```
accuracy (generic function with 2 methods)
```

```
function accuracy(ŷ, y, ϵ = 0.5)
    error = classify(ŷ, y, ϵ)
    acc = sum(error.== 1)/length(y) #accuracy
end
```

```
accuracy_white = 0.5202123315639037
```

```
accuracy_white = accuracy(ŷ_white, y1)
```

```
accuracy_red = 0.3420888055034397
```

```
accuracy_red = accuracy(ŷ_red, y2)
```

```
ϵs = 0.4:0.01:0.7
```

```
ϵs = 0.4:0.01:0.7
```

```
31
```

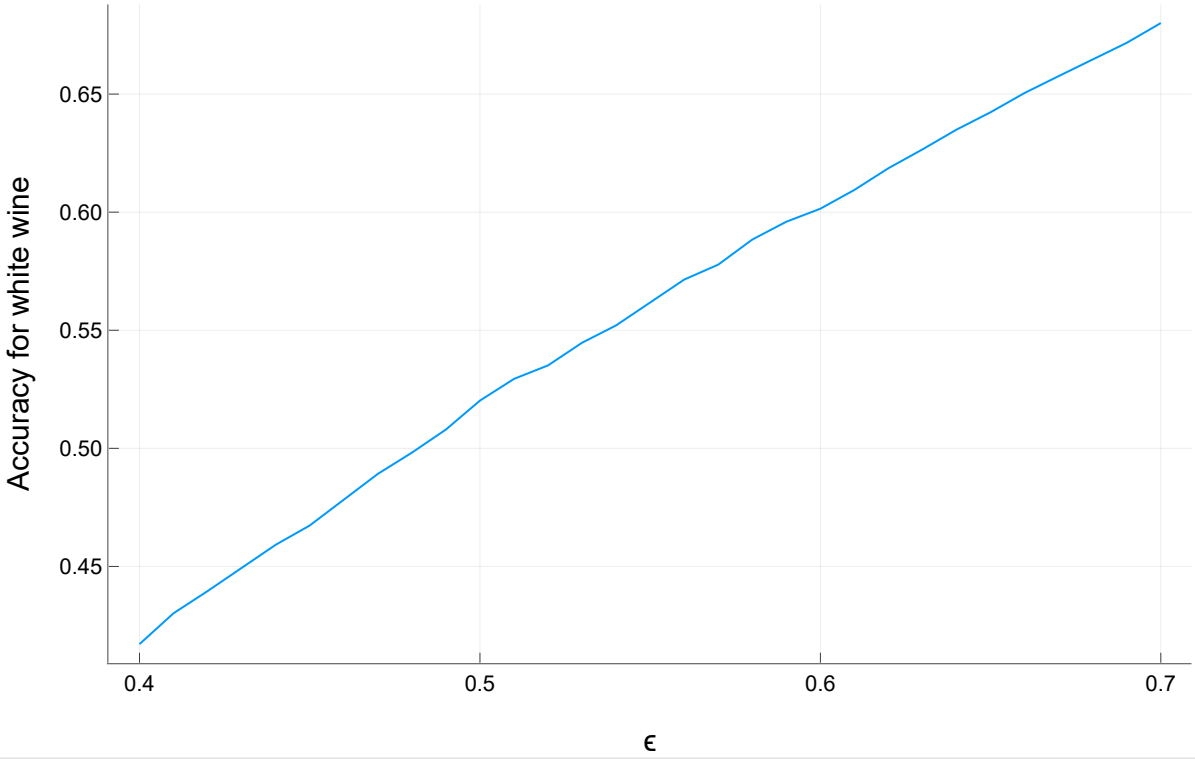
```
length(ϵs)
```

```
acc_white =
  [0.417109, 0.430176, 0.439567, 0.449367, 0.459167, 0.467334, 0.478154, 0.489179, 0.498163,
```

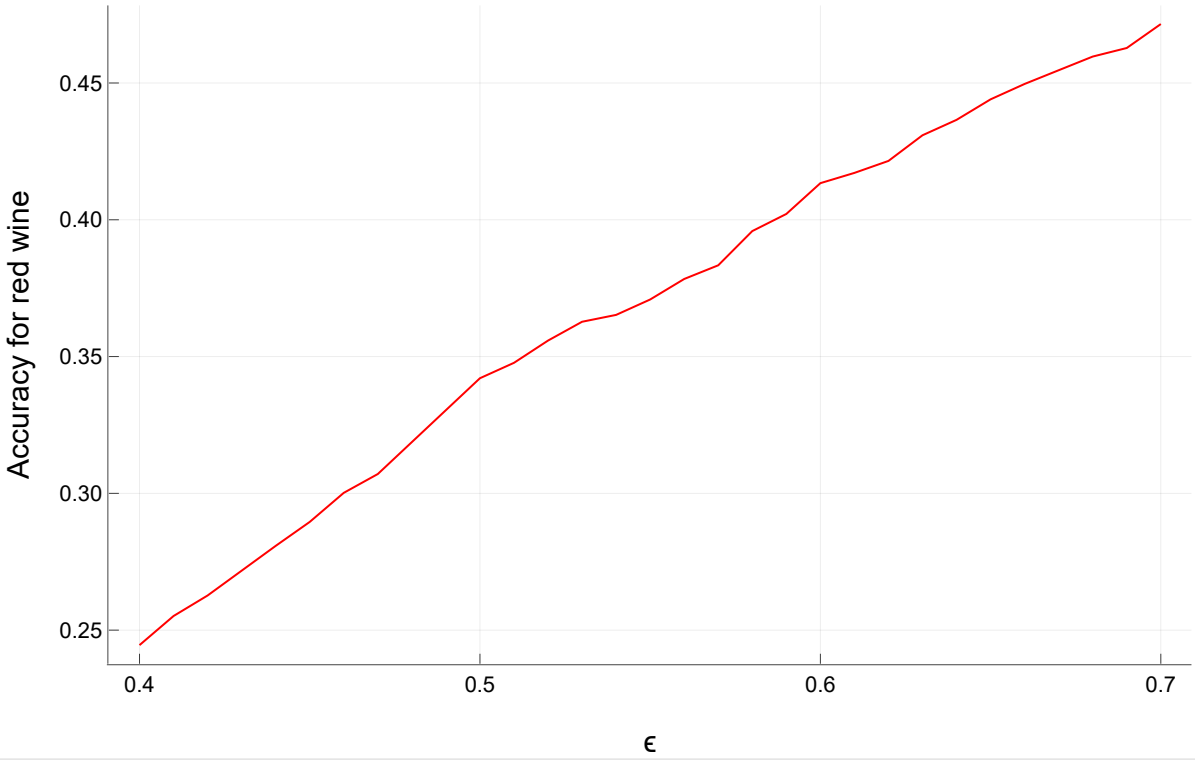
```
acc_white = [accuracy(ŷ_white, y1, ϵ) for ϵ ∈ ϵs] #ϵ
```

```
acc_red =
  [0.244528, 0.255159, 0.262664, 0.272045, 0.280801, 0.289556, 0.300188, 0.307067, 0.318949,
```

```
acc_red = [accuracy(ŷ_red, y2, ϵ) for ϵ ∈ ϵs] #ϵ
```




```
plot(ϵs, acc_white, xlabel="ϵ", ylabel="Accuracy for white wine", legend=false)
```




```
plot(ϵs, acc_red, xlabel="ϵ", ylabel="Accuracy for red wine", legend=false,
color=:red)
```