```
using DataFrames
data = readtable("medium.csv"); # pre-sorted in Excel

Q = QLearning(data);

# write policy
f = open("medium.policy","w")
for state = 1:251001
    if maximum(Q[:,state]) == 0
        #a = 10*rand(0:1);
        a = 0;
    else
        a = 10*(indmax(Q[:,state])-1);
    end
    write(f,string(a));
    write(f,"\n");
end
close(f)
```

```
function QLearning(data)
    t = 0;
    alpha = .99;
    gamma = .99;
    Q = zeros(2,251001); # (a,s) format
    Nrows = size(data,1);
    update = 0;
    repeats = 1;

    for i = 1:2000 # loop this many times, could replace with convergence check

        for line = 1:Nrows # for each row of data

            theta = round(data[line,1]*250/pi+1);
            omega = round(251+250/10*data[line,2]);
            force = data[line,3];
            thetaPrime = round(data[line,4]*250/pi+1);
            omegaPrime = round(251+250/10*data[line,5]);
            reward = data[line,6];
            st = Int(theta + (omega-1)*501); # s_t
            at = Int(force/10+1); # a_t
            rt = Int(reward);
            sp = Int(thetaPrime + (omegaPrime-1)*501);

            if line == Nrows
                stnr = Inf;
                atnr = Inf;
            else
                theta = round(data[line+1,1]*250/pi+1);
                omega = round(251+250/10*data[line+1,2]);
                force = data[line+1,3];
                stnr = theta + (omega-1)*501; # s_t next row
                atnr = force/10+1; # a_t next row
            end

            if st == stnr && at == atnr # (s,a) this row = (s,a) next row
                repeats+=1; # repeated data
                update += alpha*(rt+gamma*maximum(Q[:,sp])-Q[at,st])

            else  # next row is different
                Q[at,st] = Q[at,st] + update/repeats; # average updates based on frequencies observed in data
                update = 0;
                repeats = 1;
            end

        end
    end

    return Q
end
```