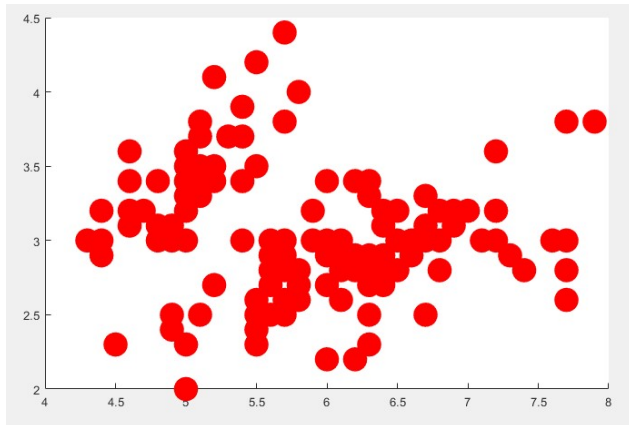
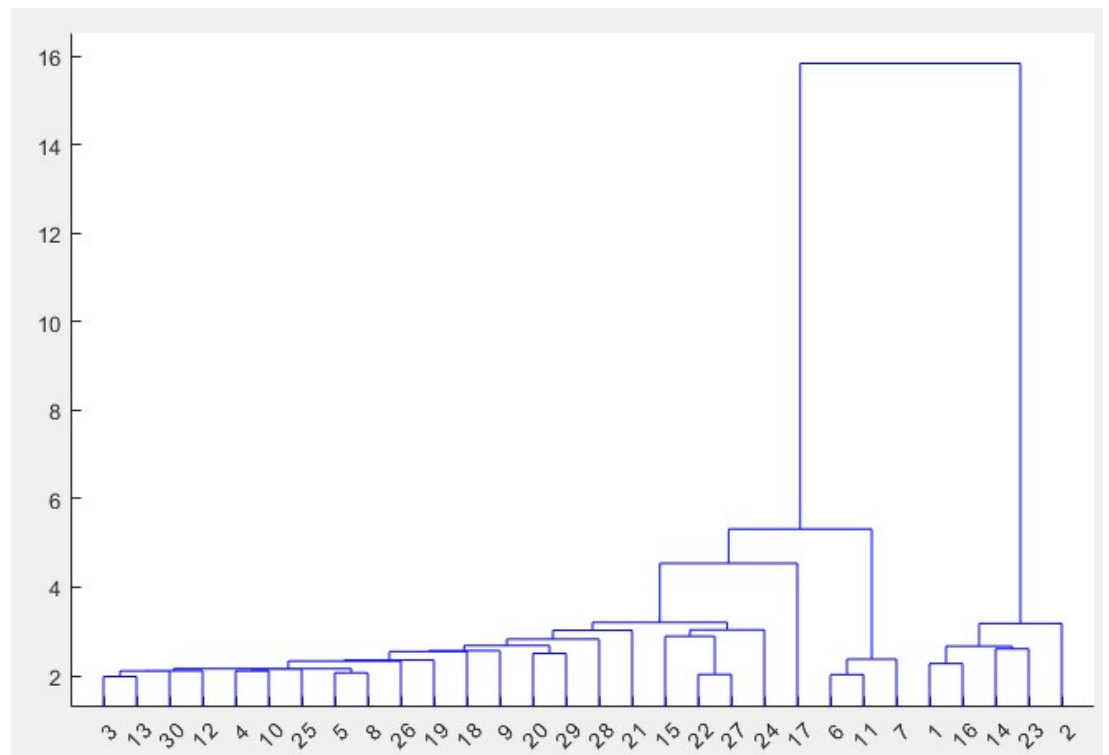


1. Cluster the iris data set using kmeans and create a dendrogram using matlab's linkage.m. Compare the results.

```
[idx,C] =  
kmeans(iris,4,'Distance','cityblock','Replicates',5,'Options',opts)  
;  
Replicate 1, 5 iterations, total sum of distances = 136.6.  
Replicate 2, 5 iterations, total sum of distances = 154.4.  
Replicate 3, 3 iterations, total sum of distances = 150.3.  
Replicate 4, 4 iterations, total sum of distances = 150.3.  
Replicate 5, 3 iterations, total sum of distances = 149.9.  
Best total sum of distances = 136.6  
>> MPD=pdist(iris);  
>> m=squareform(MPD);  
>> ml=linkage(m)
```



15.8419



2. Run **origbcm.m** on a dataset of 8 normalized random vectors. Comment on the result.

```
>> p8data=.7*ones(8)+.3*eye(8)
>> p8datan=normc(p8data)
>> p8datan'*p8datan
>> p0=initorigbcm(8,2,123)
p0 =
    struct with fields:
        wts: [1×8 double]
        rb: 0
>> pf=origbcm(p0,p8datan,100000,.005,123)
pf =
    struct with fields:
        wts: [1×8 double]
        rb: 1.0613
>> pf.wts*p8datan
ans =
    1.1530    0.7851    0.7507    0.9985    1.1826    0.8867    1.5968    1.1394
```

This is not typical because there should be one that is a lot higher than the others

```

reber.prob = [0 1 0 0 0 0 0; 0 0 0.5 0.5 0 0 0; 0 0 0.5 0 0.5 0 0; 0 0 0 0.5
0 0.5 0; 0 0 0 0.5 0 0 0.5; 0 0 0.5 0 0 0 0.5; 1 0 0 0 0 0 0]
reber.ind = [0 1 0 0 0 0 0; 0 0 2 3 0 0 0; 0 0 4 0 5 0 0; 0 0 0 4 0 5 0; 0 0
0 6 0 0 2; 0 0 6 0 0 0 3; 7 0 0 0 0 0 0]
reber.labels = 'BPQXRJE'

```

```

reber5=makestringlist(reber,4)

```

```

n0=initnet3srn(5,3,5,2,2,4312)

```

```

reber.labels(reber5.list)

```

```

stringproc(n0,[1 2 4 4 5 6 4 4 4 5 3 7],reber)

```

```

function strings = makestringlist(tgram,nstrings)
jj=1; % initial state
strings.list=[];
strings.states=[];
nstates=size(tgram.prob,2);
for ii=1:nstrings
    strings.ind(ii)=jj ; %index into superstring
    seq=[]; %initialize one string
    st=1;
    stlist=[] ; %initial state list
    while (st<nstates)
        rr=rand();
        cumu=0; i=0;
        while (cumu<rr)
            i=i+1;
            cumu=cumu+tgram.prob(st,i);
        end
        letter=tgram.ind(st,i) ;
        seq=[seq letter];
        stlist=[stlist st];
        st=i;
    end
    seq=[seq nstates] ; % append end character to seq
    strings.list=[strings.list seq];
    jj=jj+size(seq,2);
    strings.states=[strings.states stlist];
end
end

```

```

function netstruct=initnet3srn(n1,n2,n3,uamp,vamp,rs)
rng(rs);
netstruct.wih=uamp*(rand(n2,n1)-0.5) ;
netstruct.hh=uamp*(rand(n2,n2)-0.5) ;
netstruct.hbias=uamp*(rand(1,n2)-0.5) ;
netstruct.whout=vamp*(rand(n3,n2)-0.5) ;
netstruct.obias=vamp*(rand(1,n3)-0.5);
netstruct.context=zeros(1,n2);
end

```

```

function finalnet=bp3srn(net0,strlist,niter,eta,nlev)

```

```

netk=net0;
for i=1:niter
    ts=selectstring(strlist) ; % choses a new string from the training set
    netk.context=zeros(1,size(netk.wih,1)); % rests the context for a new
string
    for j=1:size(ts,2)-1 % this loop trains a single string
        netk=cyc3srn(netk,ts(j),ts(j+1),eta,nlev) ;
    end
end
finalnet=netk;
end

function [sout,hlist,slist] = stringproc(netwk,strg,gramm)
hlist=[] ;
ctxinp=zeros(1,size(netwk.wih,1));
slist=[] ;
lets=gramm.labels(strg)
%STRINGS!!!
s1=[] ;
sout=[];

for j=1:size(strg,2)-1
    hhh=hidlayersrn(strg(j),ctxinp,netwk.wih,netwk.hh,netwk.hbias,0.0);
    ou=layersig01(hhh,netwk.whout,netwk.obias);
    hlist=[hlist;hhh] ;
    s1=[s1,lets(j)];
    scell=cellstr(s1) ;
    slist=[slist;scell] ;
    sout=[sout, sprintf('%c %c',gramm.labels(strg(j)),
gramm.labels(strg(j+1)))] ;
    for kk=1:size(netwk.whout,1)
        sout=[sout sprintf('%6.3f',ou(kk))];
    end
    sout=[sout, sprintf('\n')];

    ctxinp=hhh;
end
end

```

>> Assignment8

reber =

struct with fields:

```

    prob: [7×7 double]
    ind: [7×7 double]
    labels: 'BPQXRJE'

```

reber =

struct with fields:

prob: [7×7 double]  
ind: [7×7 double]  
labels: 'BPQXRJE'

reber =

struct with fields:

prob: [7×7 double]  
ind: [7×7 double]  
labels: 'BPQXRJE'

reber5 =

struct with fields:

list: [1×73 double]  
states: [1×69 double]  
ind: [1 29 43 61]

n0 =

struct with fields:

wih: [3×5 double]  
hh: [3×3 double]  
hbias: [-0.1811 -0.6767 0.3301]  
whout: [5×3 double]  
obias: [-0.4478 0.5211 0.3970 -0.6287 0.9981]  
context: [0 0 0]

ans =

'BPRJRJXXRJRJXXRJXRJXRJXXRPEBPRJXXRJXXXRPEBPXXRJXXXXRJXRJRQEBPRJRJRJXXRQE'

lets =

'BPXXRJXXXRQE'