


```

> #problem 2
> library(plot3D)
> data = read.table(file="/Users/eotles/Downloads/PCA Service.csv",sep=",");
> cleanData = data[,2:11]
>
> pc = princomp(cleanData, cor=FALSE, center=TRUE, scale=TRUE)
> pc$loading

Loadings:
  Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
V2      -0.424  0.905
V3           0.999
V4     -0.996
V5      -0.901 -0.423
V6
V7          0.117   -0.266 -0.184 -0.936
V8          -0.129  0.952   -0.277
V9      -0.216  0.828 -0.506
V10     -0.963 -0.268
V11     -0.154  0.478  0.850  0.136

  Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
SS loadings  1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
Proportion Var 0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1
Cumulative Var 0.1    0.2    0.3    0.4    0.5    0.6    0.7    0.8    0.9    1.0
> pc$scores

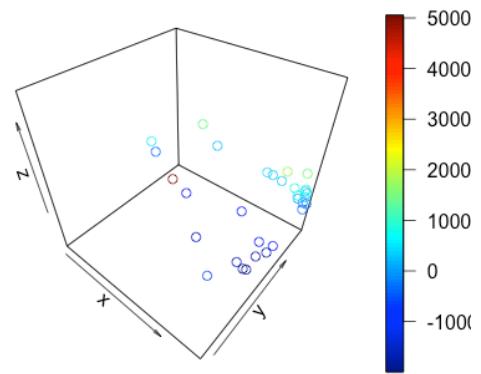
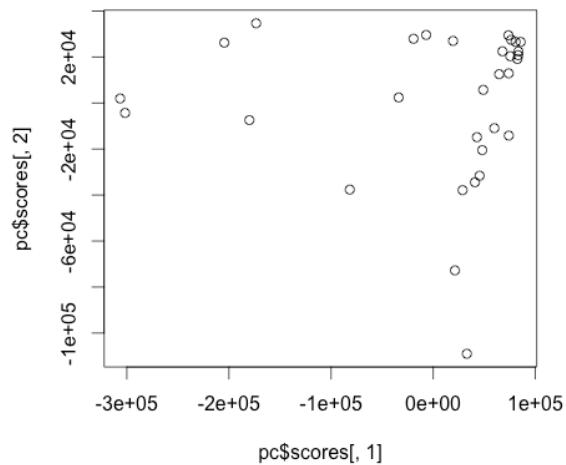
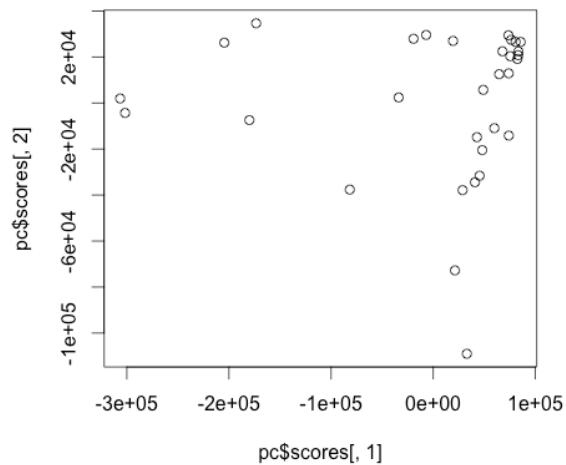
  Comp.1   Comp.2   Comp.3   Comp.4   Comp.5   Comp.6   Comp.7   Comp.8   Comp.9   Comp.10
[1,] 32899.222 -108983.817 50564.428 -145.86342 -6.0321050 6.1868250 -3.66388194 1.40990687 -2.41812141 -0.40909810
[2,] 21225.539 -72754.286 -5521.327 1001.58825 4.1566626 -4.1132154 0.07043531 -1.28678485 5.14795024 -0.96337532
[3,] 48923.784  5696.187 18351.194 221.65372 40.6696816 1.8509683 8.03330999 -0.20583644 0.01400411 0.34023671
[4,] -301725.600 -4322.373 5724.944 -130.35907 -25.0271375 5.1635592 -6.04386018 -2.93791598 1.82776790 0.27131919
[5,] 75382.208  20316.223 15432.776 -229.05705 33.5041051 -14.7512160 -2.75491774 -3.69353252 -0.34854420 0.07067969
[6,] -204605.071 26212.312 14616.237 145.64733 -34.7673899 -9.0230599 7.42411725 -0.45024246 0.08986003 0.47278613
[7,] 64542.199  12510.253 9845.799 -270.61612 13.5455808 -3.8756861 2.24546320 -0.78682011 0.29616099 1.43296640
[8,] -81759.418 -37644.170 -15307.782 -26.74775 2.6964557 -3.9965096 -1.04993256 2.28003361 1.48483368 1.12184245
[9,] 28730.843  -37888.668 -14565.891 -334.29875 7.2780914 -10.0303845 -0.60093629 1.39996290 0.74833513 -0.47338371
[10,] -306553.961 1949.018 -2247.353 -158.73064 0.9907543 -1.7755680 9.26334474 -0.89632467 -0.49218630 -1.36071552
[11,] 73830.368  12867.896 7034.582 -73.65591 -37.1424985 1.3808438 1.94925749 -0.57711898 0.70020400 2.19768083
[12,] 82360.019  19111.633 8741.682 -170.83740 -19.8397996 -5.1288202 11.04597844 -1.47304893 -0.08688129 -0.15173210
[13,] 40794.108 -34453.542 -18715.741 -1226.50318 13.2685087 -8.9323467 1.56522890 1.71216180 -0.06665505 -0.24743026
[14,] 73944.522 -14173.363 -9617.708 -670.78975 -36.9840033 21.3828951 5.93326158 0.29960465 0.35772056 -0.03859762
[15,] 42741.137 -14863.026 -11278.455 759.64978 17.0334306 3.9068726 1.34444370 -2.03768928 -0.43965274 0.58276987
[16,] 45381.009 -31615.040 -19968.300 -653.15557 7.00065864 4.9018975 -3.78941096 1.61766836 -0.18019485 0.50619838
[17,] 83199.423  20749.094 4335.543 -292.35110 -34.7570355 -22.0844830 -8.69073550 -4.38670728 0.31082716 -1.01539963
[18,] -180015.082 -7451.829 -11938.008 96.72983 22.8668450 0.3312368 1.47383182 0.14553239 -1.90590139 0.06283535
[19,] 47938.119 -20500.631 -16846.267 1009.81741 -7.4790773 -1.1272224 0.28691850 -2.64759080 -2.49003458 0.05036978
[20,] -19334.448 27907.210 4534.176 157.55240 8.9556462 -2.1192933 3.08899037 9.77518583 0.24358118 -0.88158125
[21,] 19360.472  26970.843 4090.531 30.52998 23.9289033 -0.1879204 -1.30924345 2.19536960 0.81751105 0.80702110
[22,] 80628.021  26537.347 3999.834 -182.88934 32.8640889 8.7172196 -0.75175263 -2.91618907 1.57564986 1.30573532
[23,] 67750.100  22452.357 1312.974 300.49918 -13.1328011 1.3428194 -5.83818327 4.53156963 0.46056235 -0.14303781
[24,] -6959.147 29560.480 3791.164 69.40534 50.5530863 -3.3606278 0.37587999 0.64534450 0.06748770 -0.70994557
[25,] -33895.962 2388.861 -10105.843 55.39991 -16.6896646 -5.3157304 -3.10338642 0.02580035 -1.72849794 2.99408979
[26,] -173408.754 34612.931 3402.452 -85.10099 21.3255453 13.2936611 -14.01058419 -0.18895400 -0.81863603 -0.51894168
[27,] 73561.385 29410.734 3322.037 516.77365 -21.3921424 10.8694945 -1.44603750 2.58723251 0.48065996 1.02429309
[28,] 59781.932 -10950.756 -17276.442 359.48058 0.3280593 1.8734789 1.39630686 -2.45716583 -4.01994379 -0.67029916
[29,] 83383.457 22385.435 -2830.483 -155.95561 -2.6091660 12.8210365 1.60301225 -2.85775407 0.26847005 -2.41104406

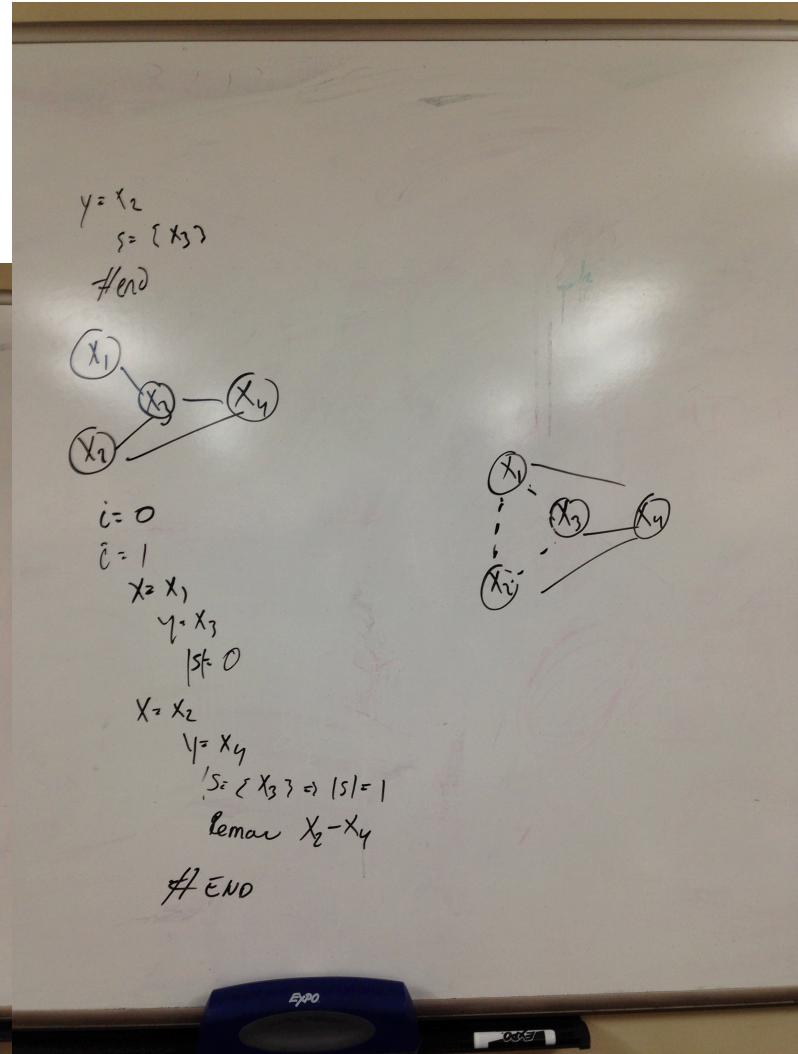
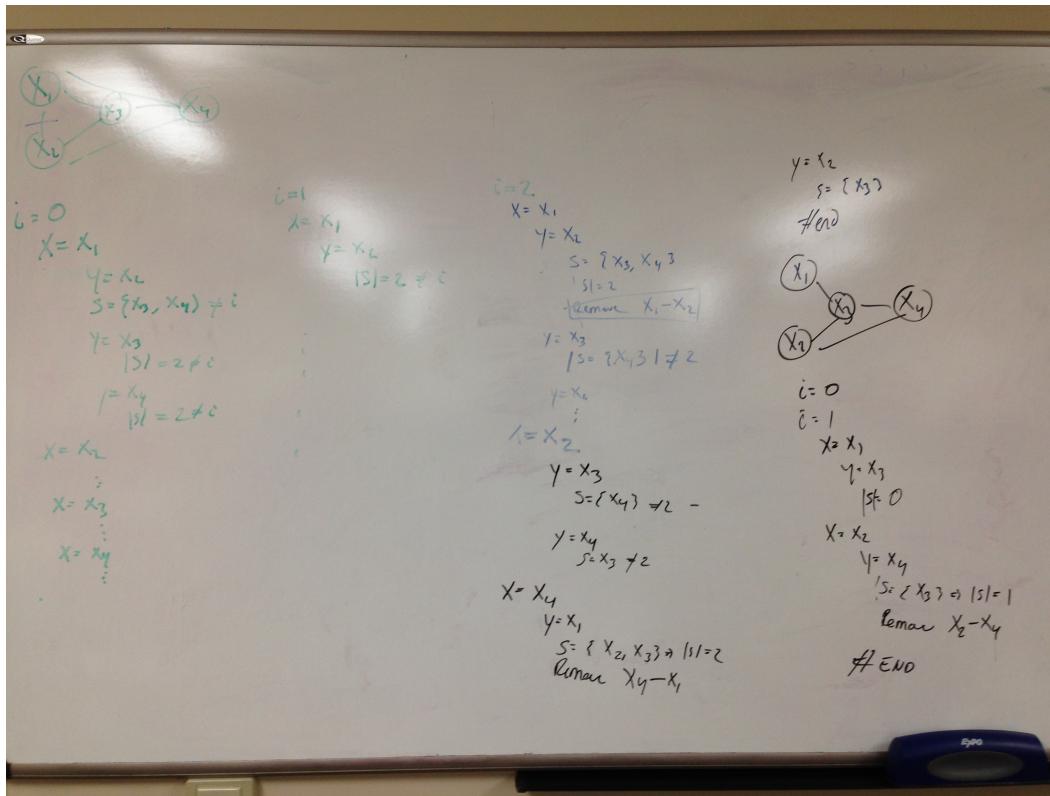
```

```

[30,] 85590.952 26515.283 -1129.316 -218.54449 -9.7200188 11.0076663 -0.50120139 -3.31573090 1.07560484 -1.43262240
[31,] 76308.622 27447.402 -1751.438 300.72878 -35.3991921 -9.2083908 -3.54571638 4.49003316 -0.97194121 -1.81361990
> plot(pc)
>
> mat_2 = pc$loading[,1:2]
> t(mat_2) %*% mat_2
   Comp.1      Comp.2
Comp.1 1.000000e+00 6.252919e-17
Comp.2 6.252919e-17 1.000000e+00
> plot(pc$scores[,1],pc$scores[,2])
>
> scatter3D(pc$scores[,1],pc$scores[,2],pc$scores[,3])

```





```

%code based on Dr. Liu's classnotes. Simply added a sample size loop.
N = 4;
dag = zeros(N,N);
C = 1; S = 2; R = 3; W = 4;
dag(C,[R S]) = 1;
dag(R,W) = 1;
dag(S,W)=1;
figure %draw the BN graph
draw_graph(dag, {'cloudy','Rain','Sprinkler','WetGrass'});

false = 1; true = 2;
ns = 2*ones(1,N); %% ns represent node_sizes, the number of values of nodes can take on
bnet = mk_bnet(dag, ns); %% Make a Bayesian network, makes a graphical model with an arc from i to j iff DAG(i,j) = 1.
bnet.CPD{C} = tabular_CPD(bnet, C, [0.5 0.5]);
bnet.CPD{R} = tabular_CPD(bnet, R, [0.8 0.2 0.2 0.8]);
bnet.CPD{S} = tabular_CPD(bnet, S, [0.5 0.9 0.5 0.1]);
bnet.CPD{W} = tabular_CPD(bnet, W, [1 0.1 0.1 0.01 0 0.9 0.9 0.99]);

sampleSize = [5,10,15,25,50,75,100,500,1000]

for ncases = sampleSize
data = zeros(N, ncases);
for m=1:ncases
    data(:,m) = cell2num(sample_bnet(bnet)); %%sample_bnet returns a Nx1 cell array, data(i,j) contains the value of i node in case j
end

%% K2
%load data.mat; % load data or use the data we simulate

C = 1; S = 2; R = 3; W = 4;

order = [C S R W]; % specify an order for the nodes

nnodes=size(data,1);

ns = 2*ones(1,nnodes); %specify the number of values for each node
max_fan_in = 2; %specify the maximum number of parents for each node

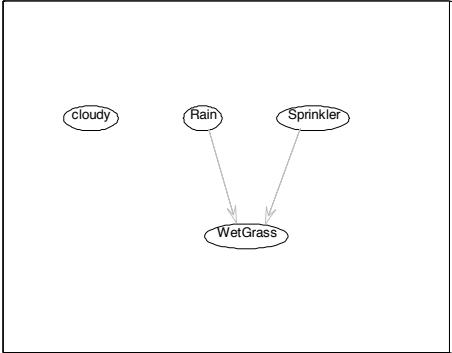
dag2 = learn_struct_K2(data, ns, order, 'max_fan_in', max_fan_in) %k2 learning

figure %draw learned BN graph
draw_graph(dag2, {'cloudy','Rain','Sprinkler','WetGrass'});

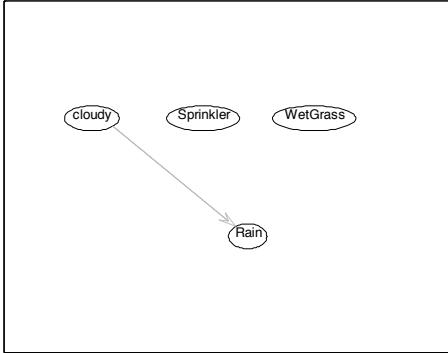
title(strcat('K2 Alogrithm ',num2str(ncases)))
end

```

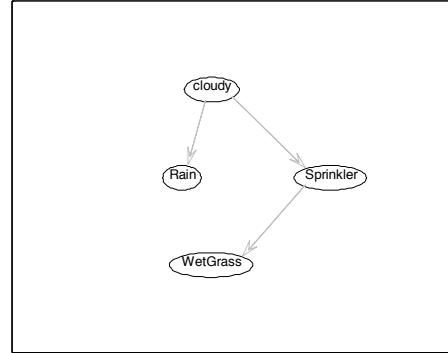
K2 Alogrithm5



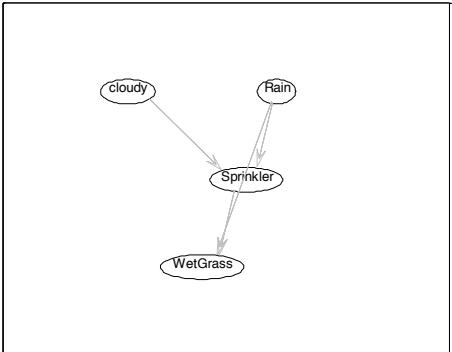
K2 Alogrithm10



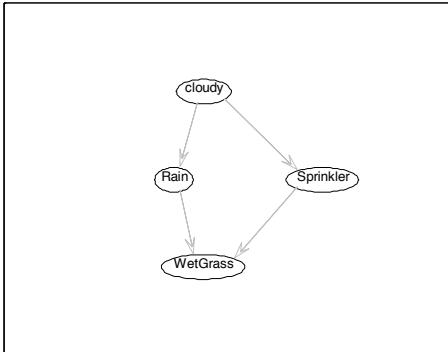
K2 Alogrithm15



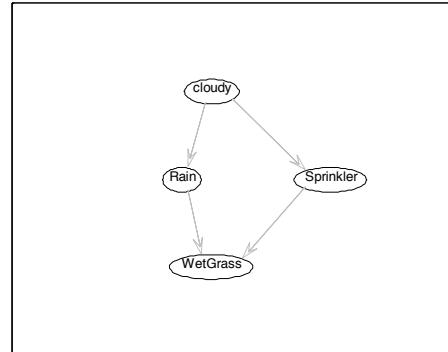
K2 Alogrithm25



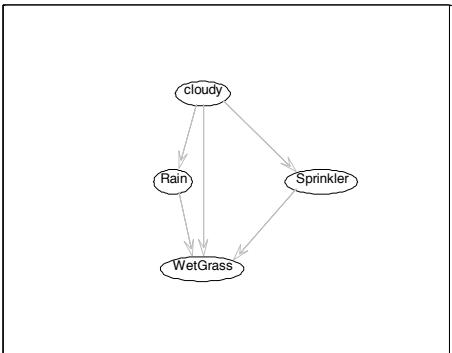
K2 Alogrithm50



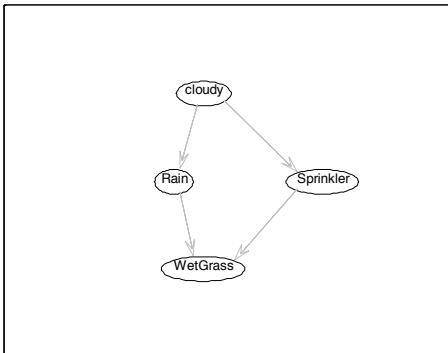
K2 Alogrithm75



K2 Alogrithm100



K2 Alogrithm500



K2 Alogrithm1000

