```
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>  setwd("C:/Users/TonyLaptop/Desktop/rowan/DM1/lecture9/lecture9")
>  data<-read.csv("censusAdult50k.csv",header = TRUE, sep = ",", stringsAsFactors = TRUE)
>  library(plyr)
> 
>  data$Race2 <- sapply(data$Race, function(b) {
+      if (b == " Black") {
+        return(1)
+      } else {
+        if (b == " White"){
+          return(0)
+      } else{
+          return(.5)
+      }}
+    }
+  )
> 
>  data$Sex2 <- sapply(data$Sex, function(b) {
+      if (b == " Female") {
+        return(1)
+      } else {
```

```
+         return(0)
+     }
+   }
+ )
>
> data$Married2 <- sapply(data$Relationship, function(b) {
+     if (b == " Husband") {
+       return(1)
+     } else {
+       if (b == " Wife"){
+         return(1)
+       }else{
+         return(0)
+   }}}
+ )
>
> minMax <- function(x,minn,maxx){
+   return((x-minn)/(maxx-minn))
+ }
> minn = min(data$Age)
> maxx = max(data$Age)
> data$Age2 <- sapply(data$Age,function(b) {
+     return((b-minn)/(maxx-minn))
+ }
+ )
>
>
>
>
> data$JobAdm2 <- sapply(data$Job, function(b) {
+     if (b == " Adm-clerical") {
+       return(1)
+     } else {
+         return(0)
+     }
+ }
+ )
>
> data$JobFarm2 <- sapply(data$Job, function(b) {
```

```
+      if (b == " Farming-fishing") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$JobOther2 <- sapply(data$Job, function(b) {
+      if (b == " Armed-Forces") {
+        return(1)
+      } else {
+        if (b == " Other"){
+          return(1)
+        }else{
+          return(0)
+      }}
+  }
+  )
>
>  data$JobCraft2 <- sapply(data$Job, function(b) {
+      if (b == " Craft-repair") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$JobExec2 <- sapply(data$Job, function(b) {
+      if (b == " Exec-managerial") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>
```

```
>
>
>
>  data$JobClean2 <- sapply(data$Job, function(b) {
+      if (b == " Handlers-cleaners") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$JobMachineOp2 <- sapply(data$Job, function(b) {
+      if (b == " Machine-op-inspct") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$JobOthServ2 <- sapply(data$Job, function(b) {
+      if (b == " Other-service") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$JobHouseServ2 <- sapply(data$Job, function(b) {
+      if (b == " Priv-house-serv") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
```

```
>  data$JobHouseServ2 <- sapply(data$Job, function(b) {
+      if (b == " Priv-house-serv") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$JobProf2 <- sapply(data$Job, function(b) {
+      if (b == " Prof-specialty") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$JobProt2 <- sapply(data$Job, function(b) {
+      if (b == " Protective-serv") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$JobSales2 <- sapply(data$Job, function(b) {
+      if (b == " Sales") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$JobTechSup2 <- sapply(data$Job, function(b) {
+      if (b == " Tech-support") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$JobMoving2 <- sapply(data$Job, function(b) {
+      if (b == " Transport-moving") {
+        return(1)
+      } else {
+          return(0)
+      }
+  }
+  )
>
>  data$Income2 <- sapply(data$Income, function(b) {
+      if (b == " <=50K") {
+        return(1)
+      } else {
+          return(0)
+    }}
+  )
>
> minn = min(data$CapitalGain)
> maxx = max(data$CapitalGain)
> data$Gain2 <- sapply(data$CapitalGain,function(b) {
+     return((b-minn)/(maxx-minn))
+ }
+ )
>
> minn = min(data$CapitalLoss)
> maxx = max(data$CapitalLoss)
> data$Loss2 <- sapply(data$CapitalLoss,function(b) {
+     return((b-minn)/(maxx-minn))
+ }
+ )
```

```
>  count(data,c('Income','Income2'))
  Income Income2  freq
1  <=50K       1 24720
2   >50K       0  7841
>  count(data,c('Sex','Sex2'))
      Sex Sex2  freq
1  Female    1 10771
2    Male    0 21790
>  count(data,c('Gain2','Loss2'))
          Gain2      Loss2  freq
1   0.000000000 0.00000000 28330
2   0.000000000 0.03558310     1
3   0.000000000 0.04889807     4
4   0.000000000 0.07415060     3
5   0.000000000 0.09618916     3
6   0.000000000 0.14348026    12
7   0.000000000 0.14990817     3
8   0.000000000 0.18595041     2
9   0.000000000 0.20202020     6
10  0.000000000 0.22359963     2
11  0.000000000 0.25068871     7
12  0.000000000 0.26124885     2
13  0.000000000 0.28879706     4
14  0.000000000 0.30762167     7
15  0.000000000 0.31680441     7
16  0.000000000 0.32323232    21
17  0.000000000 0.32392103     1
18  0.000000000 0.34090909    51
19  0.000000000 0.34527089    18
20  0.000000000 0.35330579     1
21  0.000000000 0.35904500    25
22  0.000000000 0.36111111     6
23  0.000000000 0.36248852    20
24  0.000000000 0.36501377    40
25  0.000000000 0.36593205     8
26  0.000000000 0.36776860    47
27  0.000000000 0.37121212     9
28  0.000000000 0.37373737    15
29  0.000000000 0.37832874     2
```

```
30  0.000000000 0.37901745     9
31  0.000000000 0.38292011     4
32  0.000000000 0.38314968    24
33  0.000000000 0.38383838    34
34  0.000000000 0.39462810    22
35  0.000000000 0.39508724    18
36  0.000000000 0.39623508     4
37  0.000000000 0.39830119     2
38  0.000000000 0.39944904    42
39  0.000000000 0.39967860    24
40  0.000000000 0.40289256     2
41  0.000000000 0.40449954    14
42  0.000000000 0.41689624     2
43  0.000000000 0.41896235     4
44  0.000000000 0.42332415     1
45  0.000000000 0.42424242    51
46  0.000000000 0.43067034    39
47  0.000000000 0.43319559   159
48  0.000000000 0.43663912   202
49  0.000000000 0.44628099     1
50  0.000000000 0.45316804    18
51  0.000000000 0.45385675   168
52  0.000000000 0.45454545    23
53  0.000000000 0.45936639    24
54  0.000000000 0.45959596    21
55  0.000000000 0.46877870     9
56  0.000000000 0.47084481    21
57  0.000000000 0.47222222     6
58  0.000000000 0.47750230     1
59  0.000000000 0.48875115     3
60  0.000000000 0.49334252     2
61  0.000000000 0.49655647     1
62  0.000000000 0.49908173     7
63  0.000000000 0.50022957    15
64  0.000000000 0.50528007     1
65  0.000000000 0.50619835     9
66  0.000000000 0.50642792     6
67  0.000000000 0.51216713     3
68  0.000000000 0.51377410     2
```

```
69  0.000000000 0.51561065     6
70  0.000000000 0.51836547    25
71  0.000000000 0.52043159     3
72  0.000000000 0.52387511     1
73  0.000000000 0.53696051    17
74  0.000000000 0.53994490     2
75  0.000000000 0.54568411    20
76  0.000000000 0.54912764     9
77  0.000000000 0.55440771    49
78  0.000000000 0.56106520    12
79  0.000000000 0.56404959     3
80  0.000000000 0.56634527     1
81  0.000000000 0.56749311     1
82  0.000000000 0.57139578     1
83  0.000000000 0.58471074     4
84  0.000000000 0.58746556    12
85  0.000000000 0.59756657     5
86  0.000000000 0.63223140     2
87  0.000000000 0.64830119    10
88  0.000000000 0.68962351     2
89  0.000000000 0.84550046     2
90  0.000000000 0.86547291     2
91  0.000000000 0.89531680     2
92  0.000000000 1.00000000     3
93  0.001140011 0.00000000     6
94  0.004010040 0.00000000     2
95  0.005940059 0.00000000    34
96  0.009140091 0.00000000     8
97  0.009910099 0.00000000     5
98  0.010550106 0.00000000    25
99  0.010860109 0.00000000     4
100 0.011110111 0.00000000     1
101 0.011510115 0.00000000     8
102 0.011730117 0.00000000     3
103 0.014090141 0.00000000     7
104 0.014240142 0.00000000     3
105 0.014550146 0.00000000     1
106 0.014710147 0.00000000     7
107 0.015060151 0.00000000    15
```

```
108 0.016390164 0.00000000     1
109 0.017970180 0.00000000     7
110 0.018310183 0.00000000     7
111 0.018480185 0.00000000     6
112 0.020090201 0.00000000     3
113 0.020360204 0.00000000     4
114 0.020500205 0.00000000     5
115 0.020620206 0.00000000     2
116 0.021050211 0.00000000     9
117 0.021740217 0.00000000    48
118 0.021760218 0.00000000    23
119 0.022020220 0.00000000    16
120 0.022280223 0.00000000     5
121 0.022900229 0.00000000     5
122 0.023290233 0.00000000     6
123 0.023460235 0.00000000     6
124 0.023540235 0.00000000    11
125 0.023870239 0.00000000     1
126 0.024070241 0.00000000    19
127 0.024140241 0.00000000     8
128 0.024630246 0.00000000    11
129 0.025380254 0.00000000     1
130 0.025800258 0.00000000    12
131 0.025970260 0.00000000    20
132 0.026350264 0.00000000    11
133 0.026530265 0.00000000     5
134 0.028290283 0.00000000    31
135 0.028850289 0.00000000    24
136 0.029070291 0.00000000    11
137 0.029360294 0.00000000     3
138 0.029610296 0.00000000     3
139 0.029640296 0.00000000     9
140 0.029770298 0.00000000     8
141 0.029930299 0.00000000     2
142 0.031030310 0.00000000    97
143 0.031370314 0.00000000    37
144 0.032730327 0.00000000     6
145 0.033250333 0.00000000    53
146 0.034110341 0.00000000    24
```

```
147 0.034180342 0.00000000     5
148 0.034320343 0.00000000     4
149 0.034560346 0.00000000     2
150 0.034640346 0.00000000    23
151 0.034710347 0.00000000     8
152 0.036740367 0.00000000    14
153 0.037810378 0.00000000    12
154 0.038180382 0.00000000     7
155 0.038870389 0.00000000     6
156 0.039080391 0.00000000    32
157 0.039420394 0.00000000    14
158 0.040640406 0.00000000    42
159 0.041010410 0.00000000    20
160 0.043860439 0.00000000    70
161 0.044160442 0.00000000    12
162 0.045080451 0.00000000    12
163 0.046500465 0.00000000    41
164 0.046870469 0.00000000     3
165 0.047870479 0.00000000    23
166 0.048650487 0.00000000    17
167 0.049310493 0.00000000     1
168 0.049340493 0.00000000     7
169 0.050130501 0.00000000    69
170 0.050600506 0.00000000     1
171 0.051780518 0.00000000    97
172 0.054550546 0.00000000    11
173 0.055560556 0.00000000     5
174 0.057210572 0.00000000     3
175 0.060970610 0.00000000     1
176 0.063600636 0.00000000     3
177 0.064180642 0.00000000     9
178 0.064970650 0.00000000    11
179 0.065140651 0.00000000     5
180 0.067230672 0.00000000     2
181 0.067670677 0.00000000     5
182 0.068490685 0.00000000    27
183 0.072980730 0.00000000   246
184 0.074300743 0.00000000     9
185 0.074430744 0.00000000     5
```

```
186 0.076880769 0.00000000   284
187 0.078960790 0.00000000     3
188 0.079780798 0.00000000     1
189 0.086140861 0.00000000    55
190 0.093860939 0.00000000    22
191 0.095620956 0.00000000     4
192 0.105201052 0.00000000    43
193 0.105661057 0.00000000     6
194 0.106051061 0.00000000    12
195 0.116781168 0.00000000     2
196 0.135501355 0.00000000    27
197 0.140841408 0.00000000    41
198 0.143441434 0.00000000    26
199 0.150201502 0.00000000     5
200 0.150241502 0.00000000   347
201 0.158311583 0.00000000     6
202 0.184811848 0.00000000     2
203 0.200512005 0.00000000    37
204 0.220402204 0.00000000     1
205 0.251242512 0.00000000     4
206 0.252362524 0.00000000    11
207 0.278282783 0.00000000    34
208 0.340953410 0.00000000     5
209 0.413104131 0.00000000     2
210 1.000000000 0.00000000   159
>  minn = min(data$EducationNum)
>  maxx = max(data$EducationNum)
>  data$Educ2 <- sapply(data$EducationNum,function(b) {
+      return((b-minn)/(maxx-minn))
+  }
+  )
>
>  count(data,'Educ2')
        Educ2  freq
1  0.00000000    51
2  0.06666667   168
3  0.13333333   333
4  0.20000000   646
5  0.26666667   514
```

```
6  0.33333333   933
7  0.40000000  1175
8  0.46666667   433
9  0.53333333 10501
10 0.60000000  7291
11 0.66666667  1382
12 0.73333333  1067
13 0.80000000  5355
14 0.86666667  1723
15 0.93333333   576
16 1.00000000   413
>  vars<-
c("Race2","Sex2","Married2","Age2","JobAdm2","JobFarm2","JobOther2","JobCraft2","JobExec2","JobClea
n2","JobMachineOp2","JobOthServ2","JobHouseServ2","JobProf2","JobProt2","JobSales2","JobTechSup2","
JobMoving2","Educ2","Gain2","Loss2","Income2")
>  data2<-data[vars]
>  head(data2)
  Race2 Sex2 Married2      Age2 JobAdm2 JobFarm2 JobOther2 JobCraft2 JobExec2
1     0    0        0 0.3013699       1        0         0         0        0
2     0    0        1 0.4520548       0        0         0         0        1
3     0    0        0 0.2876712       0        0         0         0        0
4     1    0        1 0.4931507       0        0         0         0        0
5     1    1        1 0.1506849       0        0         0         0        0
6     0    1        1 0.2739726       0        0         0         0        1
  JobClean2 JobMachineOp2 JobOthServ2 JobHouseServ2 JobProf2 JobProt2 JobSales2
1         0             0           0             0        0        0         0
2         0             0           0             0        0        0         0
3         1             0           0             0        0        0         0
4         1             0           0             0        0        0         0
5         0             0           0             0        1        0         0
6         0             0           0             0        0        0         0
  JobTechSup2 JobMoving2     Educ2      Gain2 Loss2 Income2
1           0          0 0.8000000 0.02174022     0       1
2           0          0 0.8000000 0.00000000     0       1
3           0          0 0.5333333 0.00000000     0       1
4           0          0 0.4000000 0.00000000     0       1
5           0          0 0.8000000 0.00000000     0       1
6           0          0 0.8666667 0.00000000     0       1
```

```
> dim(data2)
[1] 32561    22
> library("neuralnet")
Loading required package: grid
Loading required package: MASS
> set.seed(2)
> train=sample(1:nrow(data2),nrow(data2)/2)
> train=sample(1:nrow(data2),nrow(data2)/2)
> head(train)
[1] 23930 17073   146 13995 23615  6182
> test=-train
> trainingData=data2[train,]
> testingData=data2[test,]
> dim(trainingData)
[1] 16280    22
> names(trainingData)
 [1] "Race2"          "Sex2"           "Married2"       "Age2"           "JobAdm2"        "JobFarm2"
"JobOther2"      "JobCraft2"      "JobExec2"       "JobClean2"
[11] "JobMachineOp2" "JobOthServ2"   "JobHouseServ2" "JobProf2"       "JobProt2"       "JobSales2"
"JobTechSup2"    "JobMoving2"     "Educ2"          "Gain2"
[21] "Loss2"          "Income2"
> formula =
Income2~Race2+Sex2+Married2+Age2+JobAdm2+JobFarm2+JobOther2+JobCraft2+JobExec2+JobClean2+JobMachine
Op2+JobOthServ2+JobHouseServ2+JobProf2+JobProt2+JobSales2+JobTechSup2+JobMoving2+Educ2+Gain2+Loss2
> nnet<-neuralnet(formula,trainingData, hidden=20, threshold=0.4)

> #KIlled after 45 minutes; let's simplify this a bit and try again
> nnet<-neuralnet(formula,trainingData, hidden=15, threshold=0.5)
> #This took only 8 minutes; hopefully it's accurate enough
> dim(testingData)
[1] 16281    22
> names(testingData)
 [1] "Race2"          "Sex2"           "Married2"       "Age2"           "JobAdm2"        "JobFarm2"
"JobOther2"      "JobCraft2"      "JobExec2"       "JobClean2"
[11] "JobMachineOp2" "JobOthServ2"   "JobHouseServ2" "JobProf2"       "JobProt2"       "JobSales2"
"JobTechSup2"    "JobMoving2"     "Educ2"          "Gain2"
[21] "Loss2"          "Income2"
> results<-compute(nnet,testingData[,1:21])
> result<-compute(nnet,testingData[,1:21])
> testingData$result<- sapply(result$net.result, function(b) {
+    if (b<=.5){
+      return(0)
+    }else{
+      return(1)
+ }})
>  count(testingData,c('Income2','result'))
  Income2 result  freq
1       0      0  2343
2       0      1  1512
3       1      0   823
4       1      1 11603
> err=1512+823
> dim(testingData)
[1] 16281    23
> err/16281
[1] 0.1434187089
> #So even though we are probably not optimal in terms of hidden nodes we have an error rate of
14.3%, better than the 17% with decision trees and naive bayes
> #Sensitivity analysis
> library(plyr)
> trainingData$dummy<-0
> v<-ddply(trainingData, .(dummy), numcolwise(sum))
> v
  dummy  Race2 Sex2 Married2        Age2 JobAdm2 JobFarm2 JobOther2 JobCraft2 JobExec2 JobClean2
JobMachineOp2 JobOthServ2 JobHouseServ2 JobProf2 JobProt2 JobSales2
1     0 1944.5 5429     7417 4790.356164    1917      486       914      2070     2064       671
988        1631            74     2065      318      1807
  JobTechSup2 JobMoving2       Educ2       Gain2       Loss2 Income2
1         471        804 9858.866667 178.2897529 329.8227732   12294
> dim(trainingData)
[1] 16280    23
> v2<-v/16280
> v2
  dummy        Race2         Sex2     Married2         Age2      JobAdm2      JobFarm2
JobOther2    JobCraft2     JobExec2     JobClean2 JobMachineOp2
1     0 0.1194410319 0.3334766585 0.4555896806 0.2942479216 0.1177518428 0.02985257985
0.05614250614 0.1271498771 0.1267813268 0.04121621622 0.06068796069
```

```
   JobOthServ2  JobHouseServ2     JobProf2      JobProt2   JobSales2   JobTechSup2    JobMoving2
Educ2         Gain2        Loss2      Income2
1 0.1001842752 0.004545454545 0.1268427518 0.01953316953 0.110995086 0.02893120393 0.04938574939
0.6055814906 0.01095145902 0.0202593841 0.7551597052
> dim(v2)
[1]  1 23
> v3<-v2[,2:22]
> v3
         Race2         Sex2     Married2         Age2      JobAdm2      JobFarm2     JobOther2
JobCraft2     JobExec2     JobClean2 JobMachineOp2  JobOthServ2
1 0.1194410319 0.3334766585 0.4555896806 0.2942479216 0.1177518428 0.02985257985 0.05614250614
0.1271498771 0.1267813268 0.04121621622 0.06068796069 0.1001842752
   JobHouseServ2     JobProf2      JobProt2   JobSales2   JobTechSup2    JobMoving2        Educ2
Gain2        Loss2
1 0.004545454545 0.1268427518 0.01953316953 0.110995086 0.02893120393 0.04938574939 0.6055814906
0.01095145902 0.0202593841
> s1<-compute(nnet,v3)
> s1$net.result
             [,1]
[1,] 0.8795010057
> v4<-v3
> v4$Married2<-0
> s2<-compute(nnet,v4)
> s2$net.result
            [,1]
[1,] 1.015241937
> v4$Married2<-1
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.7047667191
> #So the married variable has a 30% swing from min to max
> v4<-v3
> v4<-Educ2<-0
> v4<-v3
> v4$Educ2<-0
> s2<-compute(nnet,v4)
> s2$net.result
            [,1]
[1,] 1.252284395
> v4$Educ2<-1
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.5804206563
> v4<-v3
> v4$JobTechSup2<-0
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.8451266031
> v4$JobTechSup2<-1
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.9159488836
> v4<-v3
> v4$Gain2<-0
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.8115835213
> v4$Gain2<-1
> s2<-compute(nnet,v4)
> s2$net.result
              [,1]
[1,] -0.4255341457
```

```
> v4<-v3
> v4$Loss2<-0
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.8571604304
> v4$Loss2<-1
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.1040806084
> v4<-v3
> v4$Sex2<-0
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.8966833872
> v4$Sex2<-1
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.8954283053
> v4<-v3
> v4$Race2<-0
> s2<-compute(nnet,v4)
> s2$net.result
            [,1]
[1,] 0.899016882
> v4$Race2<-1
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.7231636466
> v4<-v3
> v4$Age2<-0
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.8813848546
```

```
> v4$Age2<-1
> s2<-compute(nnet,v4)
> s2$net.result
            [,1]
[1,] 1.297283827
> v4<-v3
> v4$JobFarm2<-0
> s2<-compute(nnet,v4)
> s2$net.result
             [,1]
[1,] 0.6968951505
> v4$JobFarm2<-1
> s2<-compute(nnet,v4)
> s2$net.result
            [,1]
[1,] 1.147439974
```