Multi-dimensional scaling
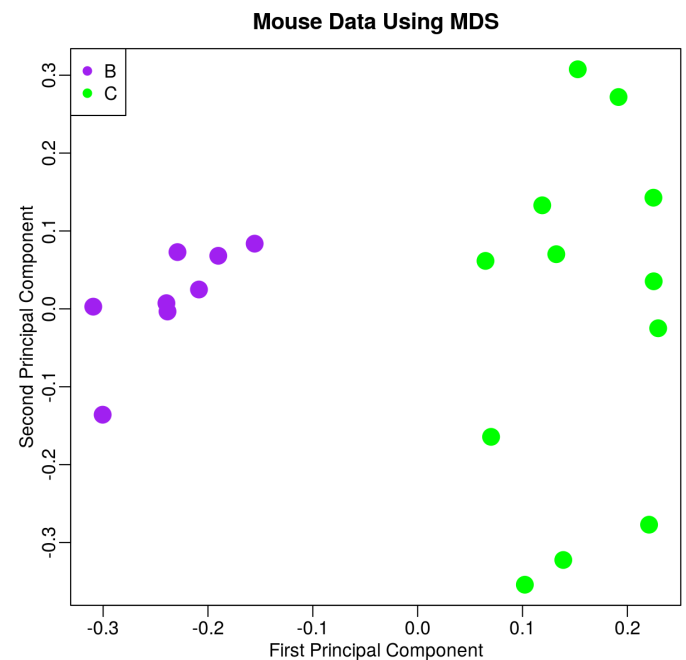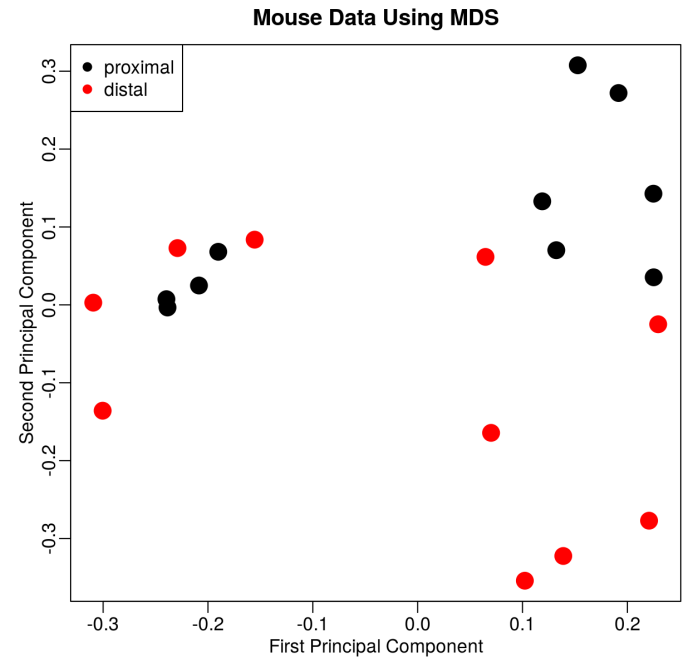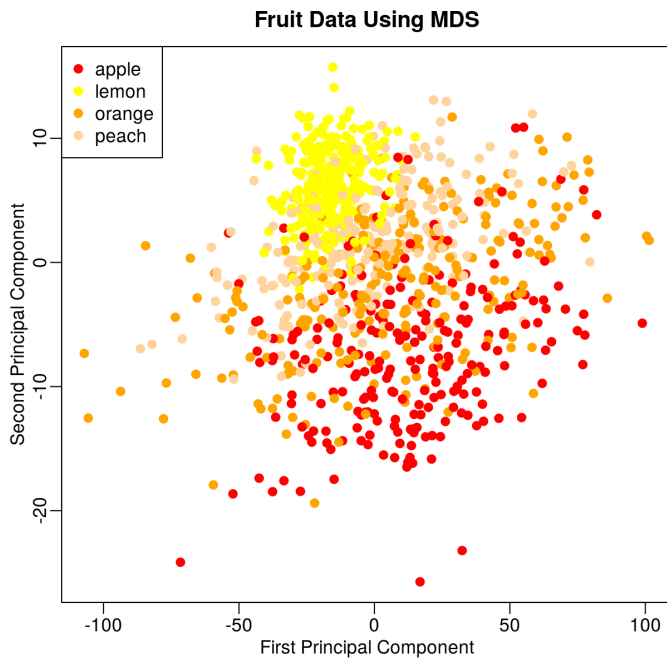
Evan Cummings

CSCI 548 – Douglas W. Raiford – Pattern Recognition

November 16, 2016

1 Fruit data MDS :

1.1 Source code :

```
# read the variable back in :
f = read.csv("../../data/fruit.csv")

# store the fruit 'classes' :
c = f[,5]

# store the fruit 'data' :
d = f[,seq(1,4)]

# perform MDS on d :
p = cmdscale(dist(d))

# store indexes of classes :
a = which(c == 'apple')
l = which(c == 'lemon')
o = which(c == 'orange')
z = which(c == 'peach')

# color vector :
cols  = c('red', 'yellow', 'orange', 'burlywood1')
cc    = as.vector(c)
cc[a] = cols[1]
cc[l] = cols[2]
cc[o] = cols[3]
cc[z] = cols[4]

# plot them colored by class :
png('../doc/images/mds_plot_fruit.png', res=200, width=6, height=6, units='in')
par(mar=c(2.5,2.5,2.5,.1),mgp = c(1.5, .5, 0))
plot(p, col=cc, bg=cc, type='p', pch=21, xlab='First Principal Component',
     ylab='Second Principal Component', main='Fruit Data Using MDS')
legend('topleft', levels(c), col = cols, pt.bg = cols, pch = 21)
dev.off()
```

2 Mouse data :

Figure 1: MDS applied to the mouse data, showing a clustering by mouse type (bottom), rather than experiment type (top).

2.1 Source code :

```
# read in the data :
f = read.csv("../../data/otu_table_L6.txt", sep="\t", row.names=1)

# get the experiment names :
e = colnames(f)

# store indexes of proximal (P) and distal (D) experiments, and mouse types
# B and C :
P = grep("[A-Z]+P[0-9]", e)
D = grep("[A-Z]+D[0-9]", e)
B = grep("^B[A-Z]+[0-9]", e)
C = grep("^C[A-Z]+[0-9]", e)

# transpose the data so each row is an experiment :
d = t(f)

# perform MDS on d :
p = cmdscale(dist(d))

# color vector :
cols  = c('black', 'red')
cc    = as.vector(e)
cc[P] = cols[1]
cc[D] = cols[2]

# plot them colored by experiment :
png('../doc/images/mds_plot_mouse_experiment.png', res=200, width=6,
    height=6, units='in')
par(mar=c(2.5,2.5,2.5,.1),mgp = c(1.5, .5, 0))
plot(p, col=cc, bg=cc, type='p', pch=21, cex=2,
     xlab='First Principal Component',
     ylab='Second Principal Component', main='Mouse Data Using MDS')
legend('topleft', c('proximal', 'distal'), col = cols, pt.bg = cols, pch = 21)
dev.off()

# color vector :
cols  = c('purple', 'green')
cc    = as.vector(e)
cc[B] = cols[1]
cc[C] = cols[2]

# plot them colored by mousey type :
png('../doc/images/mds_plot_mouse_type.png', res=200, width=6,
    height=6, units='in')
par(mar=c(2.5,2.5,2.5,.1),mgp = c(1.5, .5, 0))
plot(p, col=cc, bg=cc, type='p', pch=21, cex=2,
     xlab='First Principal Component',
     ylab='Second Principal Component', main='Mouse Data Using MDS')
legend('topleft', c('B', 'C'), col = cols, pt.bg = cols, pch = 21)
dev.off()
```

```
# color sample black for benign (b) and red for malignant (m) :
cols  = c('black', 'red')
cc    = as.vector(e)
b     = which(c == 2)
m     = which(c == 4)
cc[b] = cols[1]
cc[m] = cols[2]

# plot them colored by experiment :
png('../doc/images/mds_plot_tumor.png', res=200, width=6,
    height=6, units='in')
par(mar=c(2.5,2.5,2.5,.1),mgp = c(1.5, .5, 0))
plot(p, col=cc, bg=cc, type='p', pch=21, cex=1,
     xlab='First Principal Coordinate',
     ylab='Second Principal Coordinate', main='Tumor Data Using MDS')
legend('topleft', c('benign', 'malignant'), col = cols, pt.bg = cols, pch = 21)
dev.off()
```
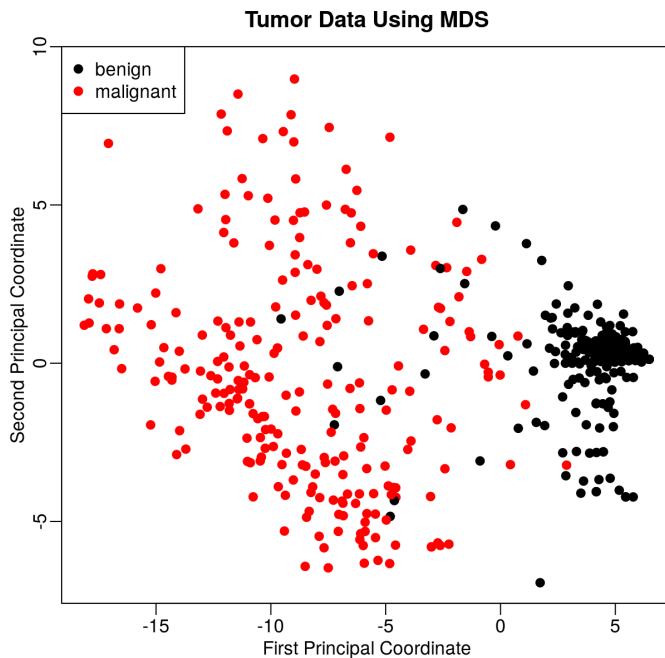
3 Tumors data :

Figure 2: MDS shows a clear separation between benign and malignant, with some overlap.

3.1 Source code :

```
# read in the data :
f = read.csv("../../data/breast-cancer-wisconsin.data", header=FALSE)

# get the sample names :
e = f[,1]

# get the classes :
c = f[,11]

# get just the data :
d = f[,seq(2,10)]

# perform MDS on d :
p = cmdscale(dist(d))
```