Math 760

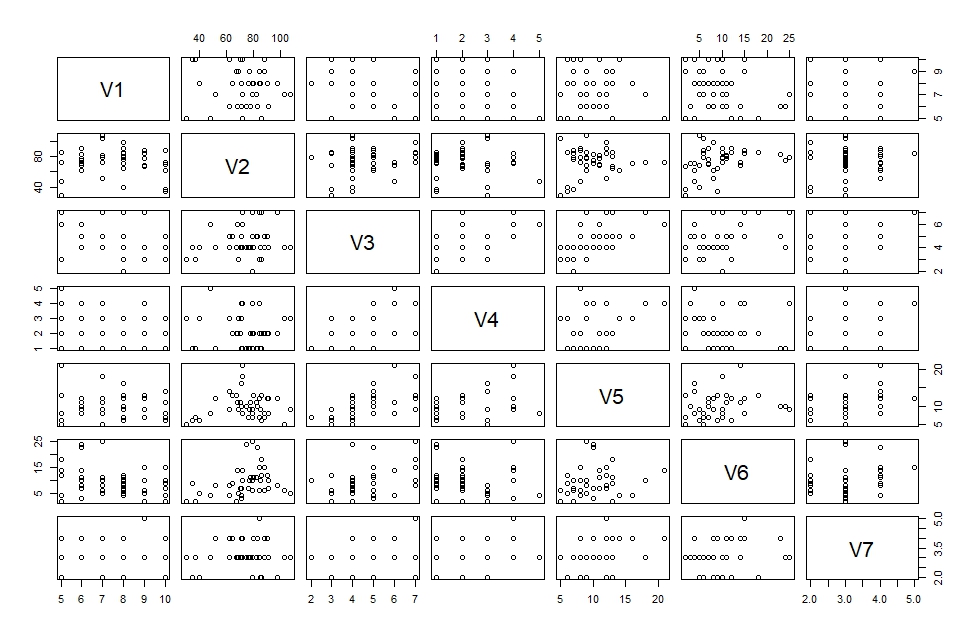
# Chapter 1 HW

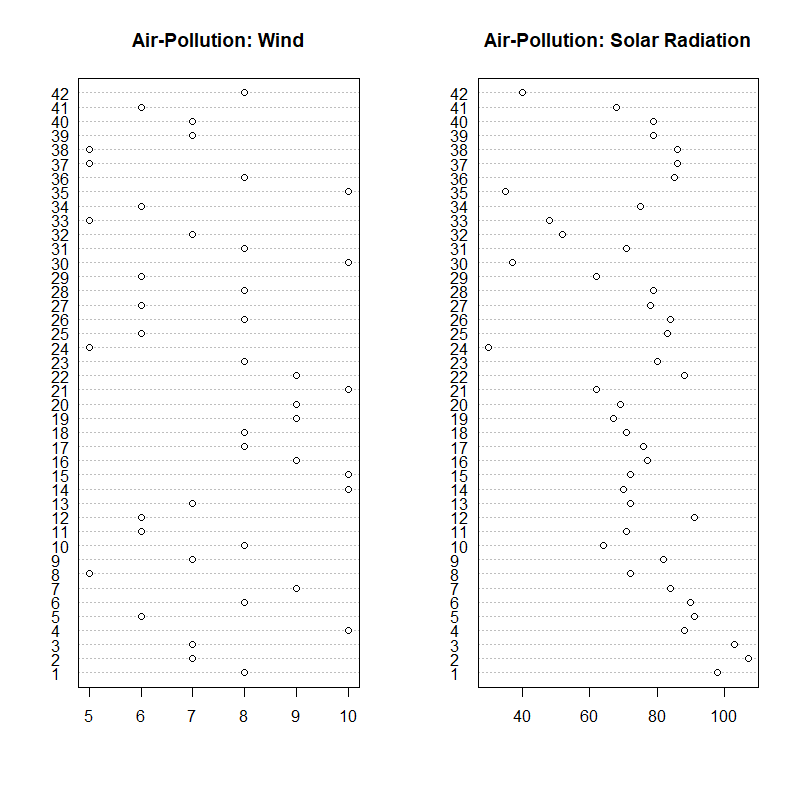
Gabrielle Salamanca

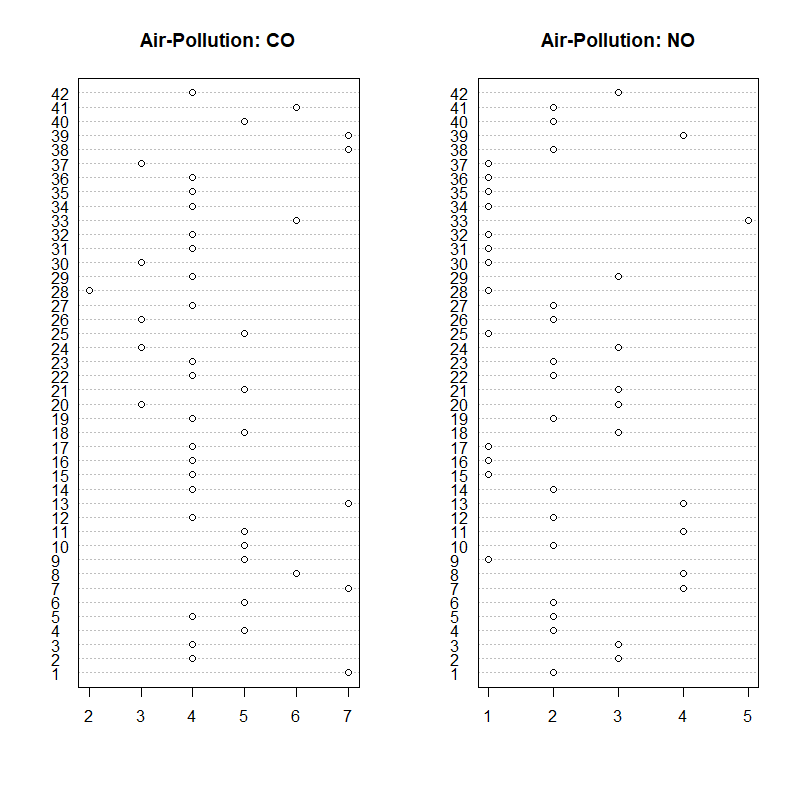
Feb 14, 2024

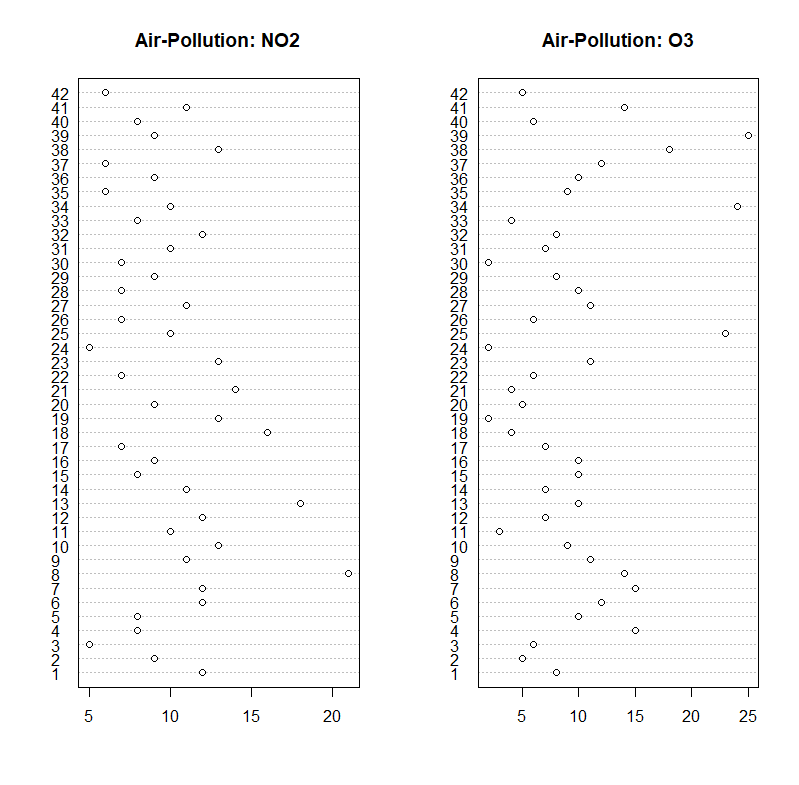
## 6. The data in Table 1.5 are 42 measurements on air-pollution variables recorded at 12:00 noon in the LA area on different days.

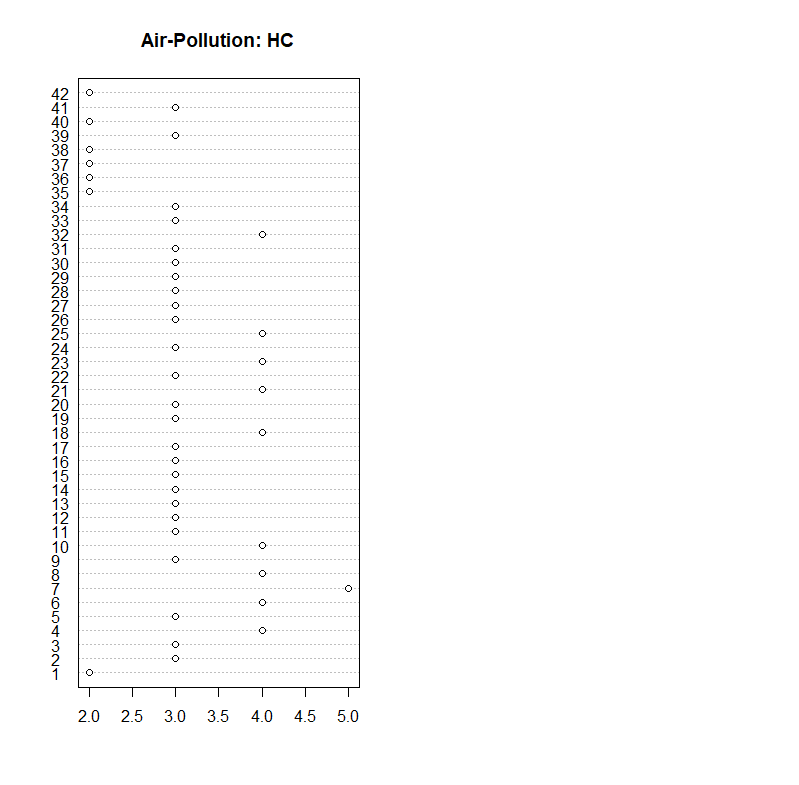
### (a) Plot the mariginal dot diagrams for all the variables.











### (b) Construct , , and R arrays, and interpret entries in R.

The array is:

## [,1]  
## V1 7.500000  
## V2 73.857143  
## V3 4.547619  
## V4 2.190476  
## V5 10.047619  
## V6 9.404762  
## V7 3.095238

The array is:

## V1 V2 V3 V4 V5 V6 V7  
## V1 2.5000000 -2.7804878 -0.3780488 -0.4634146 -0.5853659 -2.2317073 0.1707317  
## V2 -2.7804878 300.5156794 3.9094077 -1.3867596 6.7630662 30.7909408 0.6236934  
## V3 -0.3780488 3.9094077 1.5220674 0.6736353 2.3147503 2.8217189 0.1416957  
## V4 -0.4634146 -1.3867596 0.6736353 1.1823461 1.0882695 -0.8106852 0.1765389  
## V5 -0.5853659 6.7630662 2.3147503 1.0882695 11.3635308 3.1265970 1.0441347  
## V6 -2.2317073 30.7909408 2.8217189 -0.8106852 3.1265970 30.9785134 0.5946574  
## V7 0.1707317 0.6236934 0.1416957 0.1765389 1.0441347 0.5946574 0.4785134

The **R** array is:

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 1.0000000 -0.10144191 -0.1938032 -0.26954261 -0.1098249 -0.2535928  
## [2,] -0.1014419 1.00000000 0.1827934 -0.07356907 0.1157320 0.3191237  
## [3,] -0.1938032 0.18279338 1.0000000 0.50215246 0.5565838 0.4109288  
## [4,] -0.2695426 -0.07356907 0.5021525 1.00000000 0.2968981 -0.1339521  
## [5,] -0.1098249 0.11573199 0.5565838 0.29689814 1.0000000 0.1666422  
## [6,] -0.2535928 0.31912373 0.4109288 -0.13395214 0.1666422 1.0000000  
## [7,] 0.1560979 0.05201044 0.1660323 0.23470432 0.4477678 0.1544506  
## [,7]  
## [1,] 0.15609793  
## [2,] 0.05201044  
## [3,] 0.16603235  
## [4,] 0.23470432  
## [5,] 0.44776780  
## [6,] 0.15445056  
## [7,] 1.00000000

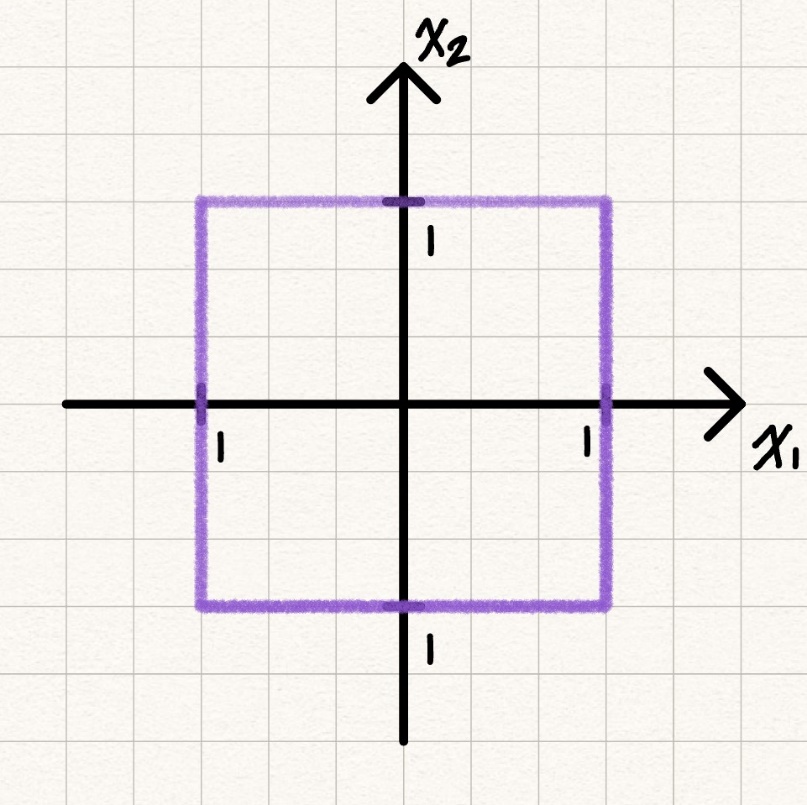
Most of the entries in the correlation array are quite small and have a negative correlation with , wind. But rows and generally have a positive correlation with the other variables, besides . Row is the only one that has a positive correlation with the rest of the variables.

## 12. Define the distance from the point to the origin as

### (a) Compute the distance from to the origin.

The distance from P = (-3,4) to the origin is 4.

### (b) Plot the locus of points whose squared distance from the origin is 1.

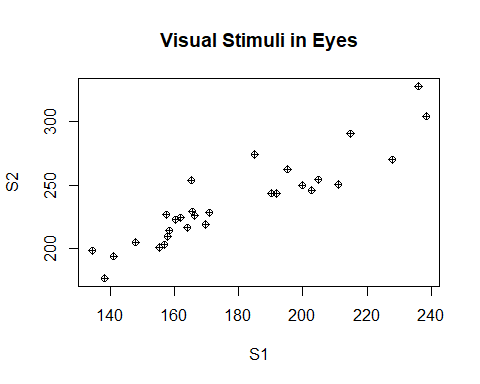


### (c) Generalize the foregoing distance expression to the points in p dimensions.

The generalization of the foregoing distance expression in terms of p dimensions is:

## 14. Table 1.6 contains some of the raw data discussed in Section 1.2. Two different visual stimuli (S1 and S2) produced responses in both the left eye (L) and the right eye (R) of subjects in the study groups. The values recoreded in the table include (subject’s age); (total response to both eyes to stimulus S1, ); (difference between responses of eyes to stimulus S1, ); and so forth.

### (a) Plot the two-dimensional scatter diagram for the variables and for multiple-sclerosis group. Comment on the appearance of the diagram.



There is a strong positive correlation according to this scatter plot, and there does not seem to be any outliers.

### (b) Compute the , , and R for the non-multiple-sclerosis and multiple-sclerosis groups separately.

Let’s start with the multiple-sclerosis group.

The array is:

## [,1]  
## V1 42.06897  
## V2 178.26897  
## V3 12.27586  
## V4 236.93103  
## V5 13.08276

The array is:

## V1 V2 V3 V4 V5  
## V1 121.13793 52.79507 -20.2197 68.13350 -29.82020  
## V2 52.79507 844.68079 244.4632 912.41493 106.76409  
## V3 -20.21970 244.46315 317.2640 232.36542 297.31921  
## V4 68.13350 912.41493 232.3654 1180.03222 81.09734  
## V5 -29.82020 106.76409 297.3192 81.09734 351.04719

The **R** array is:

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1.0000000 0.1650468 -0.1031393 0.1802078 -0.1446065  
## [2,] 0.1650468 1.0000000 0.4722334 0.9139010 0.1960632  
## [3,] -0.1031393 0.4722334 1.0000000 0.3797643 0.8909017  
## [4,] 0.1802078 0.9139010 0.3797643 1.0000000 0.1260019  
## [5,] -0.1446065 0.1960632 0.8909017 0.1260019 1.0000000

Now, let’s compute for the non-multiple-sclerosis group.

The array is:

## [,1]  
## V1 37.985507  
## V2 147.289855  
## V3 1.562319  
## V4 195.602899  
## V5 1.620290

The array is:

## V1 V2 V3 V4 V5  
## V1 277.632140 95.398380 5.361211 103.723572 3.241475  
## V2 95.398380 112.294749 1.766377 106.785030 2.042268  
## V3 5.361211 1.766377 1.805030 2.234817 0.501364  
## V4 103.723572 106.785030 2.234817 185.228815 2.351117  
## V5 3.241475 2.042268 0.501364 2.351117 2.355465

The **R** array is:

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1.0000000 0.5402894 0.2394891 0.4573917 0.1267563  
## [2,] 0.5402894 1.0000000 0.1240685 0.7404170 0.1255725  
## [3,] 0.2394891 0.1240685 1.0000000 0.1222209 0.2431491  
## [4,] 0.4573917 0.7404170 0.1222209 1.0000000 0.1125594  
## [5,] 0.1267563 0.1255725 0.2431491 0.1125594 1.0000000

## 18. Convert the national track records for women in Table 1.9 to speeds measured in meters per second. For example, the record speed for the 1100-m dash for Argentinian women is 100 m/11.57 sec 8.643 m/sec. Notice that the records for the 800-m, 1500-m, 3000-m, and marathon runs are measured in minutes. The marathon is 26.2 miles, or 42,195 meters long. Compute , , and R arrays. Notice the magnitudes of the correlation coefficients as you go from the shorter (100-meter) to the longer (marathon) running distances. Interpret these pairwise correlations.

Columns 5 through 8 will be converted from minute to second to match the first 4 columns before diving into the arrays.

The array is:

## [,1]  
## V2 11.35778  
## V3 23.11852  
## V4 51.98907  
## V5 121.34444  
## V6 251.36667  
## V7 544.84444  
## V8 9217.15556

The array is:

## V2 V3 V4 V5 V6 V7  
## V2 0.1553157 0.3445608 0.891296 1.662214 5.033472 14.03297  
## V3 0.3445608 0.8630883 2.192836 3.969954 12.165799 33.26101  
## V4 0.8912960 2.1928363 6.745458 10.908476 30.550610 85.60895  
## V5 1.6622138 3.9699539 10.908476 27.168931 77.092453 220.96553  
## V6 5.0334717 12.1657987 30.550610 77.092453 267.057736 778.15849  
## V7 14.0329686 33.2610105 85.608948 220.965535 778.158491 2393.12855  
## V8 260.0506541 623.0992537 1734.223883 4390.756730 12743.414340 38541.92805  
## V8  
## V2 260.0507  
## V3 623.0993  
## V4 1734.2239  
## V5 4390.7567  
## V6 12743.4143  
## V7 38541.9281  
## V8 972972.5414

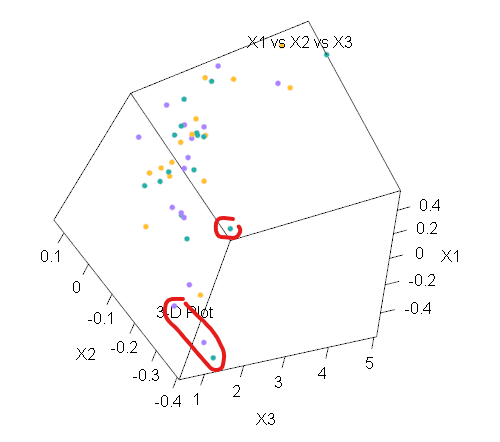
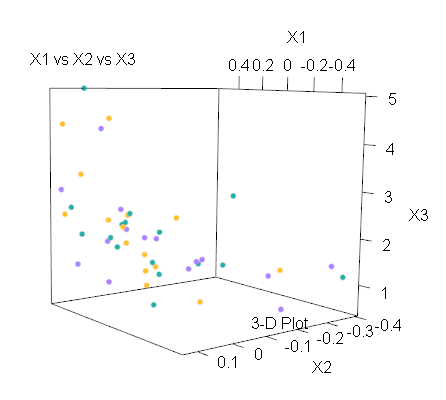
The **R** array is:

## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## [1,] 1.0000000 0.9410886 0.8707802 0.8091758 0.7815510 0.7278784 0.6689597  
## [2,] 0.9410886 1.0000000 0.9088096 0.8198258 0.8013282 0.7318546 0.6799537  
## [3,] 0.8707802 0.9088096 1.0000000 0.8057904 0.7197996 0.6737991 0.6769384  
## [4,] 0.8091758 0.8198258 0.8057904 1.0000000 0.9050509 0.8665732 0.8539900  
## [5,] 0.7815510 0.8013282 0.7197996 0.9050509 1.0000000 0.9733801 0.7905565  
## [6,] 0.7278784 0.7318546 0.6737991 0.8665732 0.9733801 1.0000000 0.7987302  
## [7,] 0.6689597 0.6799537 0.6769384 0.8539900 0.7905565 0.7987302 1.0000000

All the correlations are positive through each pair, decreasing in value as the running distances between pairs increase.

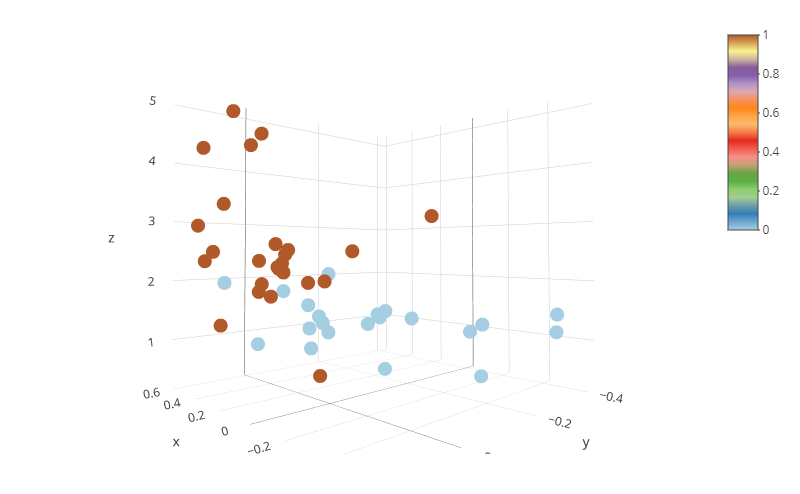
## 20. Refer to the bankruptcy data in Table 11.4, page 657, and on the folllowing website www.prenhall.com/statistics. Using appropriate computer software,

### (a) View the entire data set in , , space. Rotate the coordinate axes in various directions. Check for unusual observations.



The 3D plot shows us an exponential, whether it goes up or down depends on how you look at it. There could be a few unusual observations, ones that stray a bit far from curve. They have been circled.

### (b) Highlight the set of points corresponding to the bankrupt firms. Examine various 3D perspectives. Are there some orientations of 3D space for which bankrupt firms can be distinguished from the nonbankrupt firms? Are there observations in each of the two groups that are likely to have a significant impact on any rule developed to classify firms based on the sample means, variances, and covariances calculated from these data (See Exercise 11.24).

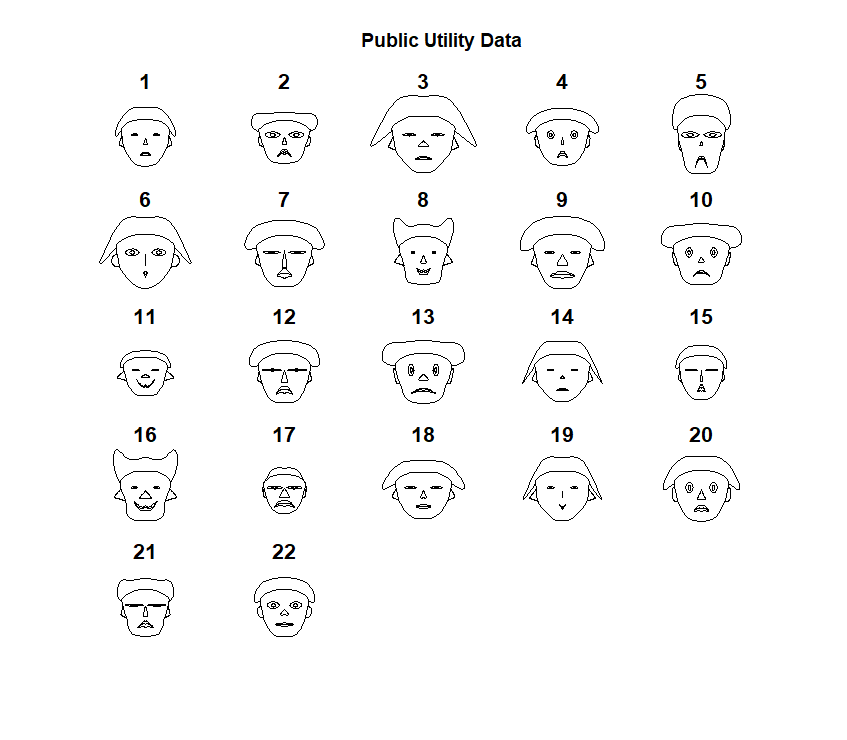


Yes, there are orientations of 3D space for which bankrupt firms can be distinguished from the non-bankrupt firms. One is shown above. They have also been color-coded to make it easier to distinguish them: the bankrupt observations are colored blue, while the non-bankrupt ones are colored brown. There are observations that are likely to have a significant impact on any rule developed to classify firms based on the sample means, variances, and covariances calculated from this data. There are a few or so observations that overlap into the group that they aren’t part of.

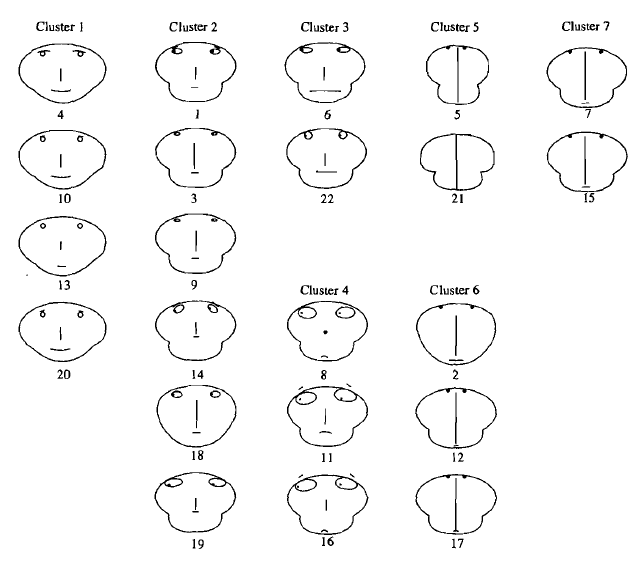
## 

## 24. Using the utility data in Table 12.4, page 688, and on the web at www.prenhall.com/statistics. represent the public utility companies as Chernoff faces with assignments of variables to facial characteristics different from those considered in Example 1.12. Compare your faces with the faces in Figure 1.17. Are different groupings indicated?

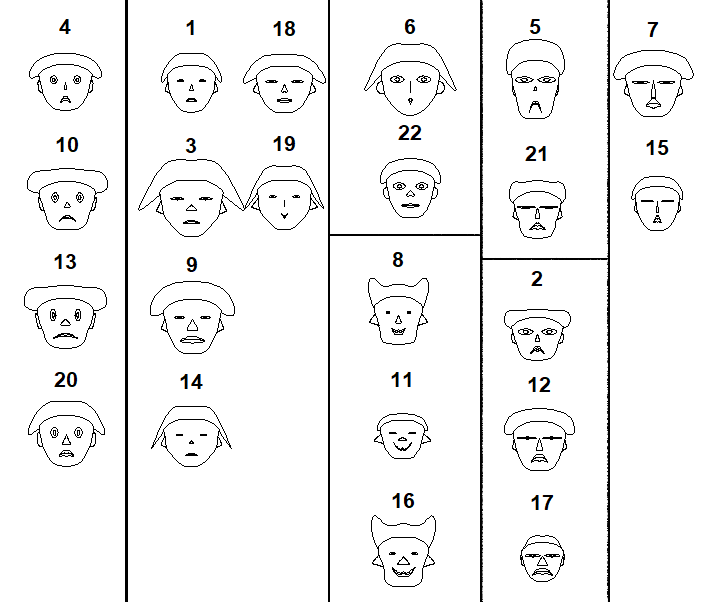
We’ll first try the Chernoff faces with the default functions.



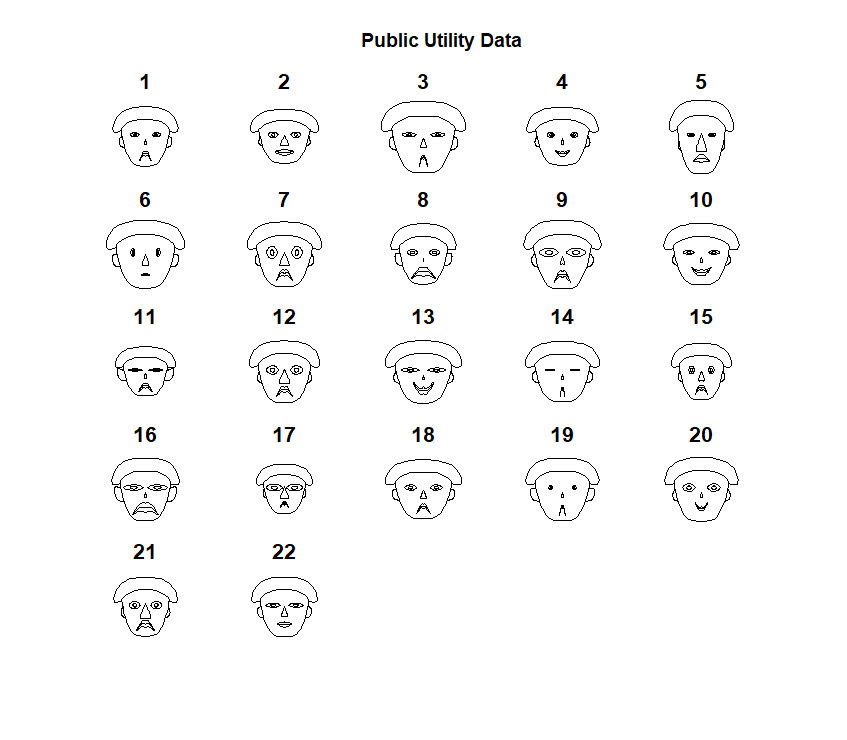
Our faces are quite notably different with the ones in Example 1.12 and Figure 1.17. There’s hair and ears on ours, while they’re lacking in Figure 1.17 below.



If we grouped our faces like in Example 1.13, I could nearly say the clusters are near similarly.



Let’s try assigning each variable with one feature, a few may double up. For the features that didn’t receive a variable, they will be assigned a constant.



Based on these new faces, the clusters could be kept the same, but I would make a few changes. I could easily group Face 11 and Face 14, both sharing closed eyes, deep frown, and small nose. Face 2 doesn’t quite fit along with Face 12 and Face 17 anymore.

# Code

knitr::opts\_chunk$set(echo = FALSE)

library(aplpack)  
library(cowplot)  
library(ggplot2)  
library(ggExtra)  
library(ggpubr)  
library(plotly)  
library(rgl)  
library(rglwidget)  
library(webshot)  
library(webshot2)  
pollute <- read.table("D:/Coding/R Storage/T1-5.dat", header = FALSE)  
  
# vars  
x1 <- pollute$V1 # wind  
x2 <- pollute$V2 # solar radiation  
x3 <- pollute$V3 # CO  
x4 <- pollute$V4 # NO  
x5 <- pollute$V5 # NO2  
x6 <- pollute$V6 # O3  
x7 <- pollute$V7 # HC  
pairs(pollute)  
  
# dot plot  
par(mfrow = c(1,2))  
dotchart(x1, labels = row.names(pollute), cex = 1, main = "Air-Pollution: Wind")  
dotchart(x2, labels = row.names(pollute), cex = 1, main = "Air-Pollution: Solar Radiation")  
dotchart(x3, labels = row.names(pollute), cex = 1, main = "Air-Pollution: CO")  
dotchart(x4, labels = row.names(pollute), cex = 1, main = "Air-Pollution: NO")  
dotchart(x5, labels = row.names(pollute), cex = 1, main = "Air-Pollution: NO2")  
dotchart(x6, labels = row.names(pollute), cex = 1, main = "Air-Pollution: O3")  
dotchart(x7, labels = row.names(pollute), cex = 1, main = "Air-Pollution: HC")  
pollMat <- as.matrix(pollute)  
ID <- as.matrix(rep(1, dim(pollMat)[1]))  
n <- dim(pollMat)[1]  
xbar <- 1/n\*t(pollMat)%\*%ID  
print(xbar)  
meanMat <- matrix(data = 1, nrow = n)%\*%cbind(xbar[[1]], xbar[[2]], xbar[[3]], xbar[[4]], xbar[[5]], xbar[[6]], xbar[[7]])  
poll <- pollMat - meanMat  
coVar <- 1/(n-1)\*t(poll)%\*%poll  
print(coVar)  
D <- diag(diag(coVar)^(-1/2))  
corr <- D%\*%coVar%\*%D  
print(corr)  
square <- ggplot() + geom\_rect(aes(xmin = -1, xmax = 1, ymin = -1, ymax = 1)) + theme\_minimal\_grid(12)  
square + coord\_equal()  
multiScler <- read.table("D:/Coding/R Storage/T1-6.dat", header = FALSE)  
  
# splitting the data by v6  
non <- subset(multiScler, V6 == 0)  
ms <- subset(multiScler, V6 == 1)  
  
# removing V6 from both sets  
non <- non[,-6]  
ms <- ms[,-6]  
# vars  
x2 <- ms$V2 #S1R + S1L  
x4 <- ms$V4 #S2R + S2L  
  
# plot  
plot(x2, x4, main = "Visual Stimuli in Eyes", xlab = "S1", ylab = "S2", pch = 10)  
msMat <- as.matrix(ms)  
ID <- as.matrix(rep(1, dim(msMat)[1]))  
n <- dim(msMat)[1]  
xbar <- 1/n\*t(msMat)%\*%ID  
print(xbar)  
meanMat <- matrix(data = 1, nrow = n)%\*%cbind(xbar[[1]], xbar[[2]], xbar[[3]], xbar[[4]], xbar[[5]])  
multi <- msMat - meanMat  
coVar <- 1/(n-1)\*t(multi)%\*%multi  
print(coVar)  
D <- diag(diag(coVar)^(-1/2))  
corr <- D%\*%coVar%\*%D  
print(corr)  
nonMat <- as.matrix(non)  
ID <- as.matrix(rep(1, dim(nonMat)[1]))  
n <- dim(nonMat)[1]  
non.xbar <- 1/n\*t(nonMat)%\*%ID  
print(non.xbar)  
non.meanMat <- matrix(data = 1, nrow = n)%\*%cbind(non.xbar[[1]], non.xbar[[2]], non.xbar[[3]], non.xbar[[4]], non.xbar[[5]])  
nons <- nonMat - non.meanMat  
non.coVar <- 1/(n-1)\*t(nons)%\*%nons  
print(non.coVar)  
nonD <- diag(diag(non.coVar)^(-1/2))  
nonCorr <- nonD%\*%non.coVar%\*%nonD  
print(nonCorr)  
track <- read.table("D:/Coding/R Storage/T1-9.dat", header = FALSE, sep = "\t")  
  
# vars  
x1 <- track$V1 # country  
x2 <- track$V2 # 100m/s  
x3 <- track$V3 # 200m/s  
x4 <- track$V4 # 400m/s  
x5 <- track$V5# 800m/min  
x6 <- track$V6 # 1500m/min  
x7 <- track$V7 # 3000m/min  
x8 <- track$V8 # marathon/min  
  
# per second  
OG <- track[,1:4]  
second <- track[,5:8]\*60  
second$V1 <- track$V1  
  
record <- merge(OG, second, by = "V1")  
record <- record[,-1]  
recordMat <- as.matrix(record)  
ID <- as.matrix(rep(1, dim(recordMat)[1]))  
n <- dim(recordMat)[1]  
xbar <- 1/n\*t(recordMat)%\*%ID  
print(xbar)  
meanMat <- matrix(data = 1, nrow = n)%\*%cbind(xbar[[1]], xbar[[2]], xbar[[3]], xbar[[4]], xbar[[5]], xbar[[6]], xbar[[7]])  
meter <- recordMat - meanMat  
coVar <- 1/(n-1)\*t(meter)%\*%meter  
print(coVar)  
D <- diag(diag(coVar)^(-1/2))  
corr <- D%\*%coVar%\*%D  
print(corr)  
bank <- read.table("D:/Coding/R Storage/T11-4.dat", header = FALSE)  
  
# vars  
x1 <- bank$V1 # CF/TD  
x2 <- bank$V2 # NI/TA  
x3 <- bank$V3 # CA/CL  
x4 <- bank$V4 # CA/NS  
x5 <- bank$V5 # pop, i = 1,2  
plot3d(x1, x2, x3, type = "p", size = 6, lit = FALSE, box = FALSE, col = c("lightseagreen","mediumpurple1","goldenrod1"),expand = 1, main = "X1 vs X2 vs X3", sub = "3-D Plot", xlab = "X1", ylab = "X2", zlab = "X3")  
plot\_ly(x = x1, y = x2, z = x3, type = "scatter3d", mode = "markers", color = x5, colors = "Paired")  
utility <- read.table("D:/Coding/R Storage/T12-4.dat", header = FALSE)  
  
# vars  
x1 <- utility$V1 # fixed-charge coverage ratio (income/debt)  
x2 <- utility$V2 # rate of return on capital  
x3 <- utility$V3 # cost per KW capacity in place  
x4 <- utility$V4 # annual load factor  
x5 <- utility$V5 # peak kWh demand growth from 1974 and 1975  
x6 <- utility$V6 # sales (kWh use/year)  
x7 <- utility$V7 # % nuclear  
x8 <- utility$V8 # total fuel costs (cents/kWh)  
x9 <- utility$V9 # company  
faces(utility[,1:8], face.type = 0, main = "Public Utility Data")  
uti <- matrix(1, nrow = 22, ncol = 15)  
  
uti[,1] <- x1  
uti[,2] <- x2  
uti[, c(4,5)] <- x3  
uti[,7] <- x4  
uti[,8] <- x5  
uti[, c(14,15)] <- x6  
uti[,6] <- x7  
uti[, c(12,13)] <- x8  
  
faces(uti, face.type = 0, main = "Public Utility Data")