Math 585 HW 4 Spring 2024 Due Friday, Feb. 23
Quiz 4 on Wed. Feb. 21, covers HW3 and 4, DD plots,
FCH, RFCH, RMVN, DGK, MB estimators, prediction regions, PCA.
3 pages, problems A)-H)

A) Below is the *R mpack* program for the DGK estimator. What is the start?

```
covdgk<-function(x, csteps = 10)
{#computes the scaled DGK multivariate estimator, need p > 1
p <- dim(x)[2]
covs <- var(x)
mns <- apply(x, 2, mean) ## concentrate
for(i in 1:csteps) {
    md2 <- mahalanobis(x, mns, covs)
    medd2 <- median(md2)
    mns <- apply(x[md2 <= medd2,  ], 2,mean)
    covs <- var(x[md2 <= medd2,  ]) }
##scale for consistency at MVN
rd2 <- mahalanobis(x, mns, covs)
const <- median(rd2)/(qchisq(0.5, p))
covs <- const * covs
list(center = mns, cov = covs)}
```

For the following problems perform the perform the *source("J:/mpack.txt")* command as described in homework 3. Also copy and paste commands from (http://parker.ad.siu.edu/Olive/mrsashw.txt) for the relevant problem into *R*.

B), 5.2 a) Download the program `ddsim`. (In *R*, type the command *library(MASS).*)

b) Using the function *ddsim* for $p = 2, 3, 4$, determine how large the sample size n should be in order for the RFCH DD plot of n $N_p(\mathbf{0}, \boldsymbol{I}_p)$ cases to cluster tightly about the identity line with high probability. Table your results. (Hint: type the command *ddsim(n=20,p=2)* and increase n by 10 until most of the 20 plots look linear. Then repeat for $p = 3$ with the n that worked for $p = 2$. Then repeat for $p = 4$ with the n that worked for $p = 3$.)

C), 5.3 a) Download the program `corrsim`. (In *R*, type the command *library(MASS).*)

b) A numerical quantity of interest is the correlation between the MD_i and RD_i in a RFCH DD plot that uses n $N_p(\mathbf{0}, \boldsymbol{I}_p)$ cases. Using the function *corrsim* for $p = 2, 3, 4$, determine how large the sample size n should be in order for 9 out of 10 correlations to be greater than 0.9. (Try to make n small.) Table your results. (Hint: type the command *corrsim(n=20,p=2,nruns=10)* and increase n by 10 until 9 or 10 of the correlations are greater than 0.9. Then repeat for $p = 3$ with the n that worked for $p = 2$. Then repeat for $p = 4$ with the n that worked for $p = 3$.)

D), 5.4 a) Download the `ddplot` function. (In *R*, type the command *library(MASS).*)

b) Using the following commands to make generate data from the EC distribution $(1-\epsilon)N_p(\mathbf{0}, \boldsymbol{I}_p) + \epsilon N_p(\mathbf{0}, 25\ \boldsymbol{I}_p)$ where $p = 3$ and $\epsilon = 0.4$.

```
n <- 400
p <- 3
eps <- 0.4
x <- matrix(rnorm(n * p), ncol = p, nrow = n)
zu <- runif(n)
x[zu < eps,] <- x[zu < eps,]*5
```

c) Use the command `ddplot(x)` to make a DD plot and include the plot in *Word.* What is the slope of the line followed by the plotted points? (Right click *Stop* once on the plot.)

E), 5.5 a) Download the `ellipse` function.

b) Use the following commands to create a bivariate data set with outliers and to obtain a classical and robust RMVN covering ellipsoid. Include the two plots in *Word.*

```
simx2 <- matrix(rnorm(200),nrow=100,ncol=2)
outx2 <- matrix(10 + rnorm(80),nrow=40,ncol=2)
outx2 <- rbind(outx2,simx2)
ellipse(outx2)

zout <- covrmvn(outx2)
ellipse(outx2,center=zout$center,cov=zout$cov)
```

F), 4.8 The *mpack* function `covesim` compares various ways to robustly estimate the covariance matrix. The estimators used are ccov: the classical estimator applied to the clean cases, RFCH and RMVN. The average dispersion matrix is reported over nruns = 20. Let diag(A) be the diagonal of the average dispersion matrix. Then diagdiff = diag(ccov) - diag(rmvne) and abssumd = sum(abs(diagdiff)). The clean data $N_p(0, diag(1, ..., p))$.

a) The *R* command covesim(n=100,p=4) gives output when there are no outliers. Copy and paste the output into *Word.*

b) The command covesim(n=100,p=4,outliers=1,pm=15) uses 40% outliers that are a tight cluster at major axis with mean $(0, ..., 0, pm)^T$. Hence *pm* determines how far the outliers are from the bulk of the data. Copy and paste the output into *Word.* The average dispersion matrices should be $\approx c\ diag(1, 2, 3, 4)$ for this type of outlier configuration. What is c for RFCH and RMVN?

G), 5.8 Use the *R* command *source("J:/mrobdata.txt")* then *ddplot4(buxx,alpha=0.2)* and put the plot in *Word*. The Buxton data has 5 outliers, $p = 4$, and $n = 87$, so the 80% prediction regions use $1 - \alpha + p/n = 0.846$ percentiles. The output shows that the cutoffs are 2.527, 2.734 and 2.583 for the nonparametric, semiparametric and robust parametric prediction regions. The two horizontal lines that correspond to the robust distances are obscured by the identity line.

H) Shown below is PCA output using the correlation matrix for the Buxton data where 5 outliers were deleted. The variables were *length, nasal height, bigonal breadth, cephalic* and $buxy = height/20$. The "standard deviations" line corresponds to the square roots of the eigenvalues. The Rotation matrix gives the 5 principal components.

a) For the robust `rprcomp` output make a scree plot. What proportion of the trace is explained by the first 4 principal components?

b) Which principal component corresponds to i) bigonal, ii) nasal + buxy, iii) length + cephalic, iv) length − cephalic and v) nasal − buxy?

```
rprcomp(z)
$out
Standard deviations:
[1] 1.3369152 1.1466891 1.0016463 0.8123854 0.4842482

Rotation:
                 PC1         PC2         PC3         PC4         PC5
len       0.67271620 -0.21639022  0.05559575  0.15178244 -0.68883916
nasal    -0.22213361 -0.66957907  0.05173705 -0.68978370 -0.15440936
bigonal  -0.01373814  0.02995162  0.99668240  0.03545927  0.06542933
cephalic -0.67269993  0.21806615  0.02362841  0.16076405 -0.68812686
buxy     -0.21306252 -0.67556583 -0.01727087  0.68851877  0.15446292

prcomp(z,scale=T)
Standard deviations:
[1] 1.3184358 1.1723991 1.0155266 0.7867349 0.4867867

Rotation:
                 PC1         PC2        PC3         PC4        PC5
len      -0.70308364 -0.06777853 0.07743938  0.16900791  0.6830219
nasal    -0.15038248  0.68867720 0.02042098 -0.70384733  0.0853859
bigonal  -0.11646120 -0.04882199 0.96504341 -0.02261327 -0.2285455
cephalic  0.68502160  0.08950469 0.24854103  0.03070660  0.6782468
buxy     -0.01551443  0.71465734 0.02246533  0.68889840 -0.1180614
```