Homework Assignment # 2
Due Date: March 28, 2016

Instructor: Daniel P. Gaile, PhD
STA 525: Statistics For Bioinformatics

March 8, 2016

Problem 1: Applying PCA to example datasets

1. Perform Principal Components Analyses of the Berry, 2010
   (a) Each group will be assigned it's own dataset and subset of factors to examine.
      i. Group 1:

```
# Berry UK Test and Training (array)
gse19435=getGEO("GSE19435",GSEMatrix=T)
show(gse19435)
pdat19435=pData(phenoData(gse19435[[1]]))[,c(1,11:16)]
```

      ii. Group 2:

```
# Berry UK Training (array)
gse19439=getGEO("GSE19439",GSEMatrix=T)
show(gse19439)
pdat19439=pData(phenoData(gse19439[[1]]))[,c(1,11:16)]
```

      iii. Group 3:

```
 # Berry UK Test (array)
gse19444=getGEO("GSE19444",GSEMatrix=T)
show(gse19444)
pdat19444=pData(phenoData(gse19444[[1]]))[,c(1,11:16)]
```

      iv. Group 4:

```
# Berry South Africa (array)
gse19442=getGEO("GSE19442",GSEMatrix=T)
show(gse19442)
pdat19442=pData(phenoData(gse19442[[1]]))[,c(1,11:16)]
```

   (b) Go to Geo PubMed and look up the information for your group's data-set.
   (c) Perform PCA and investigate whether or not any of the factor levels (for each of the factors assigned to your group) can be adequately discriminated using the first two principal components.
   (d) Implement a feature selection step and perform PCA on the feature selected data-set.

Problem 2: Applying hierarchical clustering to example datasets

1. Clustering Exercise
   (a) Load your group's assigned Berry, 2010 dataset.
   (b) Perform clustering (of samples) using hclust. and the full data. Test for association to the categorical clinical values that you have for the subjects.
      i. Try a few different distance metrics
      ii. Try a few different clustering methods
      iii. Report and compare your results
   (c) Repeat the analysis for feature selected subsets of your Berry, 2010.
      i. Report and compare your results to those using the full data.
      ii. What are your conclusions regarding the feature selection step that you used for your analysis?

Problem 3: Power

In this problem we explore the concept of power in the context of a simplified multiple testing scenario. Specifically, we consider a biomarker study involving the comparison of 200 targets (e.g., gene expression probesets) across two patient populations (i.e., "group A" and "group B", each with $n = 10$ subjects) with the goal of identifying targets that are differentiately "expressed" across the two groups. We will assume that only one of the targets is differentially expressed while the balance (i.e., 199) are not. We will assume that the target expression values are distributed normally with mean 0 and standard deviation 1 for the 199 "null" targets and with mean δ and standard deviation 1 for the single "alternative" target. We will assume that the null targets have a multivariate distribution with a pairwise correlation of ρ for all null target pairings.

Let's assume that our analysis plan for the data will be to perform two sample equal variance t-tests and then use maxT to adjust the p-values. Why might we prefer the maxT adjustment to that of Bonferroni?

One quick way to estimate the power of our proposed study and analysis is to bound it between two power curves. Let's first consider the single truly differentiately expressed target. If we had only tested (with a parametric equal variance t-test) that particular target, what would be the power to reject $H_0 : \mu_A = \mu_B$ if $\alpha = 0.05$ and $\mu_A = 0$ and $\mu_B = \delta$? We can estimate the power using the pwr.t.test function from the pwr library:

```
  require(pwr)
```

```
## Loading required package:  pwr
```

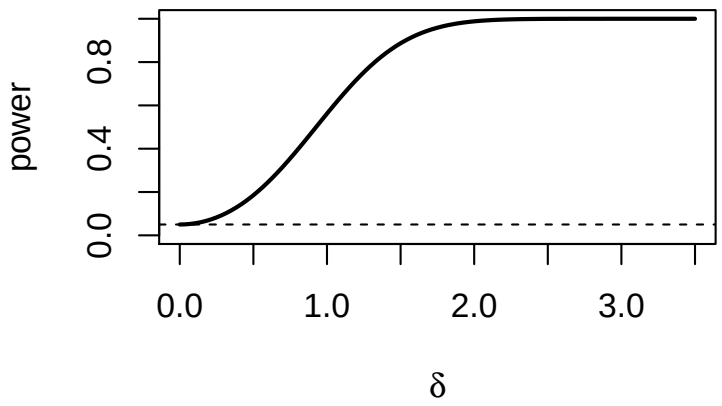
```
 # calculate power for n=10, sig.level=0.05, and d=1

  pwr.t.test(n=10,d=1,sig.level=0.05)
```

```
##
##      Two-sample t test power calculation
##
##               n = 10
##               d = 1
##       sig.level = 0.05
##           power = 0.5620066
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
 # what if we wanted to know the minimum detectable effect size for 0.80 power

  pwr.t.test(n=10,sig.level=0.05,power=0.8)
```

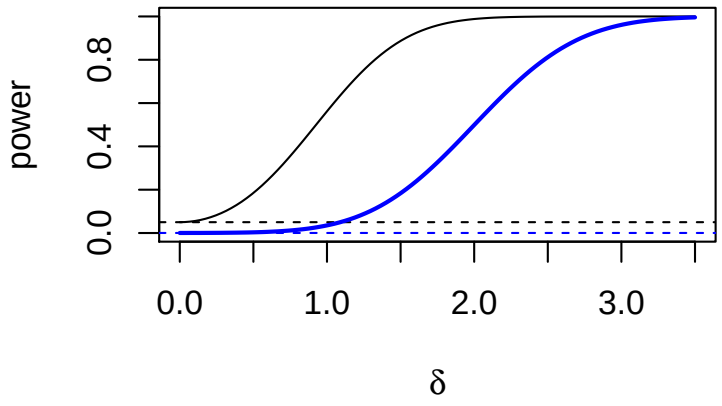
```
##
##      Two-sample t test power calculation
##
##               n = 10
##               d = 1.324947
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
 # now, let's calculate power for a collection of d values

  dvals=seq(0,3.5,length=101)
  pwr=rep(0,length(dvals))

  for(i in 1:(length(dvals))) pwr[i]=pwr.t.test(n=10,d=dvals[i],sig.level=0.05)$power

  # make figure
  plot(dvals,pwr,type="l",xlab=expression(delta),ylab="power",lwd=2,ylim=c(0,1))
  abline(h=0.05,lty=2)
```



If the test of the truly differentially expressed target was conducted as a part of the set of 200 tests across all features then we could control the family-wise error rate (FWER) using a Bonferroni correction. Namely, we could conduct each of the 200 tests at a level of $\alpha = 0.05/200$ and the FWER will be controlled at a value no greater than 0.05. What would be the power to reject the null hypothesis if the test was conducted at the Bonferroni corrected level of $\alpha = 0.05/200$?

```
pwrBon=rep(0,length(dvals))

for(i in 1:(length(dvals))) pwrBon[i]=pwr.t.test(n=10,d=dvals[i],sig.level=0.05/200)$power

# make figure
plot(dvals,pwr,type="l",xlab=expression(delta),ylab="power",ylim=c(0,1))
abline(h=0.05,lty=2)
lines(dvals,pwrBon,col=4,lwd=2)
abline(h=0.05/200,lty=2,col=4)
```




```
# black curve corresponds to alpha=0.05 and blue curve corresponds to alpha=0.05/200
```

So what would the power curve look like if we adjusted for test multiplicity using maxT instead of Bonferroni? Remember that the Bonferroni correction is reasonable if the set of tests are independent but it can be very conservative if the tests are correlated. The maxT approach, on the other hand, can adjust for the multiplicity while

accounting for the underyling correlation structure. Therefore, it is reasonable to expect that the power curve for a maxT approach will lie somewhere between the two power curves in the previous figure.

For this homework problem you will use the following function:

```
  #============================================================#
#
# Simple Multivariate Normal Simulation
#
#============================================================#

# A simulation function to generate a matrix of data:
# the simulated data will be a matrix that has:
#
#   2*n columns with the first n columns corresponding to n samples from a 'control' group
#               and the second n columns correspnding to n samples from a 'treatment' group
#
#   p.alt+p.null rows with the first p.alt rows corresponding to the the p.alt features (e.g. genes)
#                      that are diferrentially expressed and the last p.null rows corresponding to
#                      p.null features that aren't differentiall expressed
#
#  the matrix values will all be generated from a multivariate normal distribution with covariance
#  matrix:
#          |     1    rho.alt rho.alt ...       0       0       0    |
#          |  rho.alt    1    rho.alt ...       0       0       0    |
#          |     :       :       :                      :       :    |
#          |     0       0       0    ...       1    rho.nul rho.nul |
#          |     0       0       0    ...    rho.nul    1    rho.nul |
#          |     0       0       0    ...    rho.nul rho.nul    1    |
#
# The p.null rows will all have mean zero while the p.alt rows will have mean zero for
# the first n columns and mean delta for the last n columns
# (i.e., the columns corresponding to 'treatment').
#
# Additionally, the function allows for sample specific centering error
#  from a normal distribution with mean 0 and sd = sdC


SimDatHW2=function(p.alt=10, # the number of differentially expressed features
                   p.null=90, # the number of non-differentially expressed features
                   n=20, # the number of samples in each of the treatment and control groups
                   rho.alt=0.2, # correlation of the alt hyp variables
                   rho.null=0.1, # correlation of the null hyp variables
                   delta=2,# the mean of the p.alt features in the "treatment" group
                   sdC=0 # the variance of the sample specific centering error
){

  p=p.alt+p.null
  Sigma=array(rep(0,p^2),dim=c(p,p))
  Sigma[1:p.alt,1:p.alt]=rho.alt
  Sigma[(p.alt+(1:p.null)),(p.alt+(1:p.null))]=rho.null
  diag(Sigma)=1
  Xc=mvrnorm(n,mu=rep(0,p),Sigma=Sigma)
  Xt=mvrnorm(n,mu=c(rep(delta,p.alt),rep(0,p.null)),Sigma=Sigma)
  x=t(rbind(Xc,Xt))
  colnames(x)=c(paste("cntrl",1:n,sep=""),paste("trt",1:n,sep=""))
  rownames(x)=c(paste("DEgene",1:(p.alt)),paste("nonDEgene",1:p.null,sep=""))
  if(sdC>0){
    CentError=rnorm(2*n,sd=sdC)
    for(j in 1:(2*n)) x[,j]=x[,j]+CentError[j]
  }
```

```
  return(x)
}
```

For example, we can run the following simulation:

```
if(F){ # I ran this as if(T) the first time, then changed it
       # so I don't have to rerun it every time
  nreps=1000
  PhiVec=rep(NA,nreps)

  for(k in 1:nreps){

    X=SimDatHW2(p.alt=1,p.null=199,n=10,rho.alt=0.0,rho.null=0.75,delta=2)
    # calculate minP adjusted p-values
    resT <- mt.maxT(X, f, B = 2500)
    PhiVec[k]=as.integer(resT$adjp[resT$index==1]<=0.05)
  }
  save(file="PhiVec.RData",PhiVec)
} # end if(F)

  load(file="PhiVec.RData") # provides PhiVec
```

```
plot(dvals,pwr,type="l",xlab=expression(delta),ylab="power",ylim=c(0,1))
abline(h=0.05,lty=2)
lines(dvals,pwrBon,col=4,lwd=2)
abline(h=0.05/200,lty=2,col=4)
# add our simulation point
points(2,mean(PhiVec),pch="+",col="darkgreen")
```

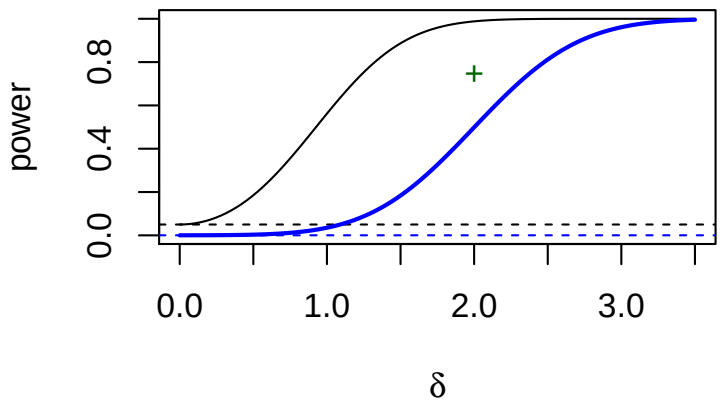


Figure 1: Power curves: Level $\alpha = 0.05$ (black) and level $\alpha = 0.05/200$ (blue). The plotted green point corresponds to our simulated estimate of power when using maxT adjustment instead of Bonferroni.

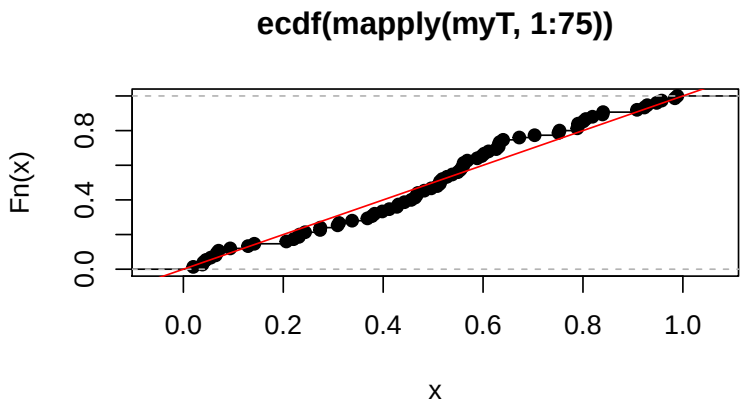
Run a simulation and add some more points to Figure 1. Specifically, perform simulations for $\rho \in \{0.25, 0.75\}$ and $\delta \in \{0.75, 1.50, 2.25, 3.0\}$. Interpret your results. What can you say about the gains in power when using maxT instead of Bonferroni? Do the results for different values of ρ make sense?

Problem 4:

Centering Errors Can Induce Spurious Correlations

In this problem we are going to consider a small array dataset and explore the effects that centering error can have on estimation and inference. We begin by generating a small array dataset using the simulation function in Problem 2.

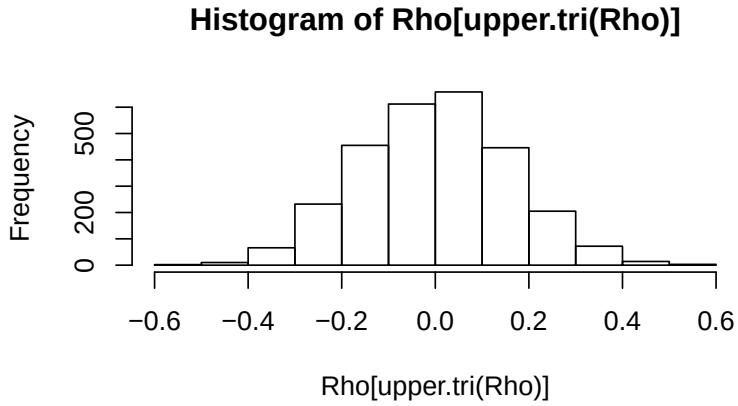
```
# note with delta=0, all 75 assays are technically null
X= SimDatHW2(p.alt=25,p.null=50,n=20,rho.alt=0.0,rho.null=0.0,delta=0,sdC=0)

# Let's look at the ecdf of the p-values (should be uniform)
myT=function(i) t.test(X[i,1:20],X[i,21:40],var.equal=T)$p.value
plot(ecdf(mapply(myT,1:75)));abline(0,1,col=2)
```



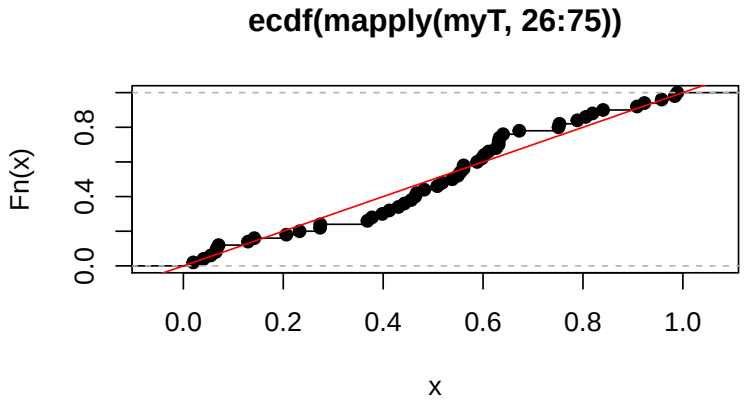

```
# and, indeed, they look pretty uniform

# Let's look at all the pairwise correlations between targets:
# how many are there?
choose(75,2)
```

```
## [1] 2775
```

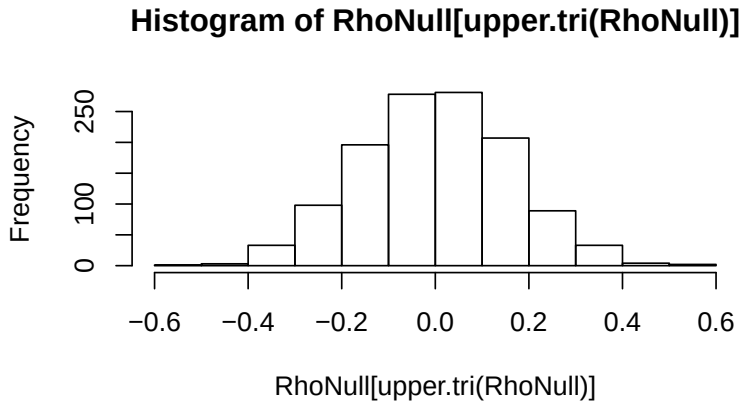
```
# [1] 2775

# here is an easy way to get those values:
Rho=cor(t(X))
hist(Rho[upper.tri(Rho)])
```



```
# note how the distribution is centered about 0.0, as we would expect since we generated
# all the observations independently.

# suppose we wanted to just examine just the null targets
# (you want to do this for the datasets mentioned below
#     when delta and rho.alt are both not 0)

# here is how to code that in an easy fashion..
plot(ecdf(mapply(myT,26:75)));abline(0,1,col=2)
```




```
# and...
RhoNull=cor(t(X[26:75,]))
hist(RhoNull[upper.tri(RhoNull)])
```



For this problem, you will investigate the p-value and correlation distributions for the following three simulated datasets:

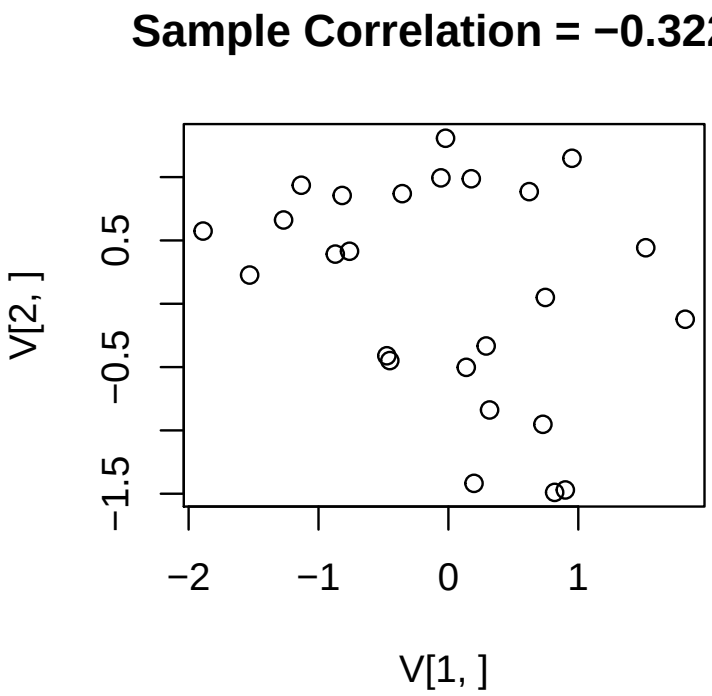
```
# Dataset 1:
#     generated with no centering/batch errors.
#     correlation among the "alternative" targets but none among the nulls
X1= SimDatHW2(p.alt=25,p.null=50,n=20,rho.alt=.3,rho.null=0.0,delta=2,sdC=0)

# Dataset 2a:
#     generated with centering/batch errors.
#     correlation among the "alternative" targets but none among the nulls

X2a=  SimDatHW2(p.alt=25,p.null=50,n=20,rho.alt=.3,rho.null=0.0,delta=2,sdC=1)
```

```r
# Dataset 2b:
#     Dataset 2a but re-centered about each sample (i.e., column) mean.
#     Mean centering of this type is not an uncommon practice among small panel arrays

X2b= X2a
for(j in 1:40) X2b[,j]=X2b[,j]-mean(X2b[,j])
```

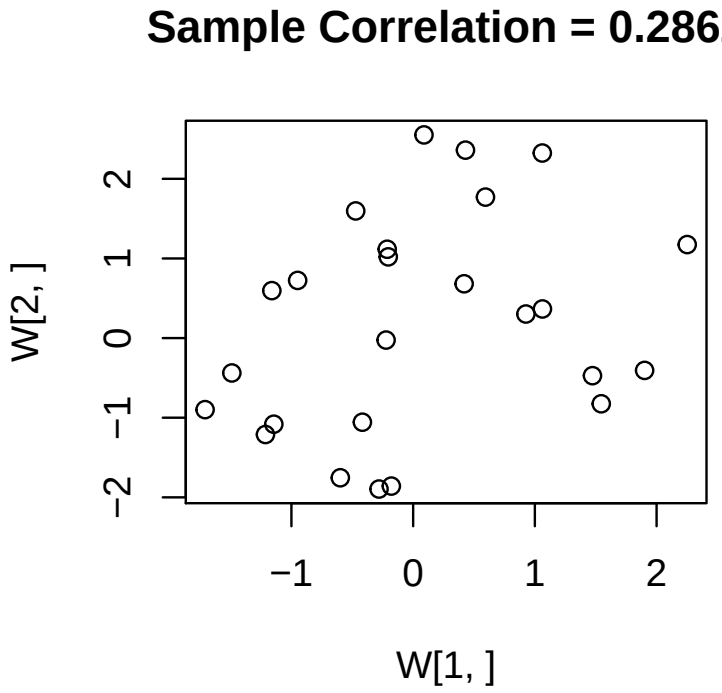
When you examine the p-values and correlation estimates, make sure to do so according to whether the targets are alt or null. Note, that for each of the three datasets, the null targets were generated with zero correlation, so one might expect their pairwise correlation distributions to be centered about zero as in our first example. Is that the case? Additionally, all the nulls were generated such that the p-values for those targets should have a uniform distn. Is that the case? Interpret your results. Can centering errors and other uncorrectable batch effects cause problems with estimation and inference?

If you are struggling to understand what happened, consider this as a hint:

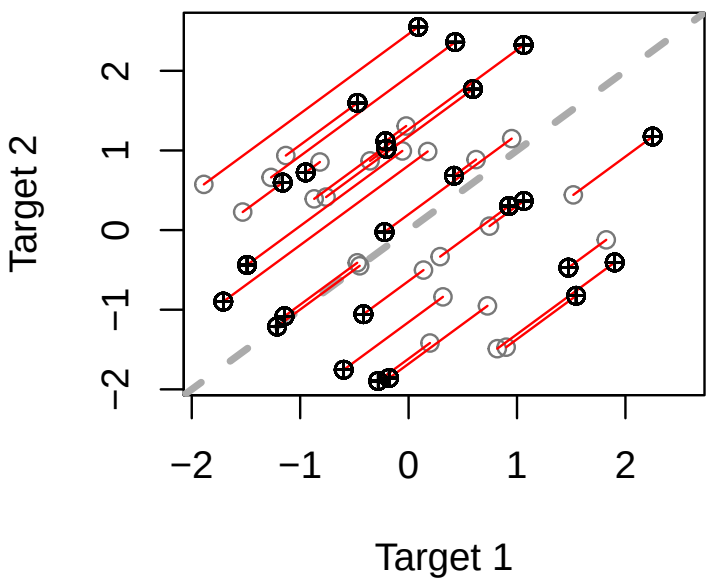
```r
# data generated, all independent, no centering error
V=matrix(rnorm(50),ncol=25)
# let's look at the plot
plot(V[1,],V[2,],main=paste("Sample Correlation =",round(cor(V[1,],V[2,]),4)))
```




```r
# now, let's add centering error to each column
W=V
for(j in 1:25) W[,j]=W[,j]+rnorm(n=1)
# let's look at the plot
plot(W[1,],W[2,],main=paste("Sample Correlation =",round(cor(W[1,],W[2,]),4)))
```




```
# what happened ?

# Let's make a more illustrative plot
xylim=range(c(V[1,],W[1,],V[2,],W[2,]))
plot(c(V[1,],W[1,]),c(V[2,],W[2,]),
      xlim=xylim,ylim=xylim,type="n",xlab="Target 1", ylab="Target 2")
abline(0,1,lty=2,lwd=3,col="gray67")
# now,let's add original points
points(V[1,],V[2,],col="gray47")
# now, let's add the points with centering error and connect with lines
for(j in 1:25){
  lines(c(V[1,j],W[1,j]),c(V[2,j],W[2,j]),col=2)
  points(W[1,],W[2,],pch=10)
}
```




```
# note that all the red lines have slope equal to 1. Why is that?
```

Problem 5: The Mystery Matrix

I have uploaded the file "HW3X.RData" to the Homework 3 folder. The file contains a column-wise scrambled data matrix, X. Which is to say, that there is an orginal data matrix, Y, that has a coherant structure. I simply took the columns of Y and randomly shuffled them to generate X. Using data analytic techniques/heuristics applied only to some or all of the data in the matrix X, can you figure out a way to restore the proper ordering of the columns of X?

For this problem explore a variety of clustering (and perhaps clustering related techniques). Provide a description in your write-up of the types of approaches that you considered. Then, provide the code and column sorted heatmap image for the approach that you felt provided you the best results.

I toyed around with this problem for an hour or so the other night and I was able to restore a large portion of the proper ordering. So, it can be done!

```
load("HW2X.RData")
```

```
## two plots side by side (option fig.show='hold')
heatmap(X,Rowv=NA,Colv=NA)
```

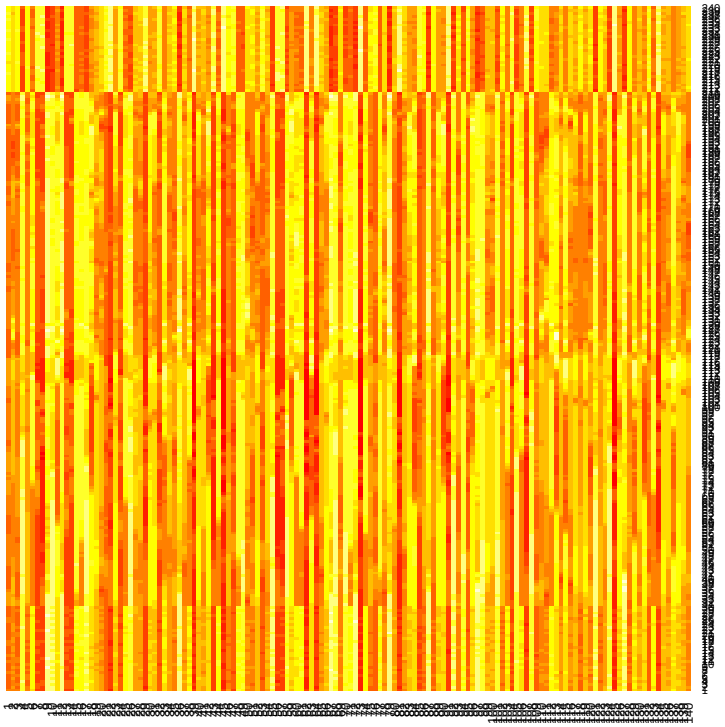


Figure 2: The column shuffled data matrix. I used the command: heatmap(X,Rowv=NA,ColV=NA) to generate this figure. In your homework solution, use the command heatmap(X[,ordDX],Rowv=NA,ColV=NA), where ordDX is the re-ordering of the columns that you arrived at.