Software: Multivariate t-test (parametric and by permutation)


```
# paired Hotelling's T^2: Erik Otárola-Castillo 2/14/2012

library(MASS)
##create a multivariate group with a mean of 0, and sd =1
group1 <- cbind(rnorm(10, mean = 0, sd = 1), rnorm(10, mean = 0, sd = 1),rnorm(10, mean = 0, sd = 1));group1

##create a multivariate group with a mean of 2, and sd =1
group2 <- cbind(rnorm(10, mean = 2, sd = 1), rnorm(10, mean = 2, sd = 1),rnorm(10, mean = 2, sd = 1));group2

# sample difference
numbers<-group1-group2;numbers

# mean vector of the sample difference
dbar <- apply(numbers, 2, mean);dbar

# n # of rows
n <- nrow(numbers);n

# k # of columns
k <- ncol(numbers);k

# variance covariance matrix of the difference matrix (VCV)
S <- cov(numbers);S

# invert the VCV (S^-1) using the generalized inverse in MASS (ginv)
S.inv<-ginv(S);S.inv

# transpose dbar dbar^t
dbar.t<-t(dbar);dbar.t

# calculate t^2 value, in matrix notation: n*dbar^t * S^-1 * dbar
t2<-n*(dbar.t%*%S.inv%*%dbar);t2

# transform T^2 to F ratio , this is the "observed" F in non-parametric permutation below
F <- ((n-k)/((n-1)*k))*t2;F

# calculate parametric p-value from an f distribution using F-ratio with k and n-k degrees of freedom
P <- 1-pf(F,k,n-k);P
```

```
# Now let's get a p value BY PERMUTATION :-) it is much more powerful

# the question becomes: "can we get a greater F ratio by random chance alone?
# number of permutations
nperm<-9999
sig<-numeric(nperm)
F.permb<-numeric(nperm)
# group two groups into one matrix
gmatrix<-rbind(group1,group2)

for (i in 1:nperm)  {
cat("Perm No.", i,"\n")
  # sample difference
  g1.perm<-gmatrix[sample(nrow(gmatrix),nrow(group1)),]
  g2.perm<-gmatrix[sample(nrow(gmatrix),nrow(group2)),]
  numbers.perm<-g1.perm-g2.perm
  # mean vector of the sample difference
  dbar.perm <- apply(numbers.perm, 2, mean)
  # n # of rows
  n.perm <- nrow(numbers.perm)
  # k # of columns
  k.perm <- ncol(numbers.perm)
  # variance covariance matrix of the difference matrix (VCV)
  S.perm <- cov(numbers.perm)
  # invert the VCV (S^-1) using the generalized inverse in MASS (ginv)
  S.inv.perm<-ginv(S.perm)
  # transpose dbar dbar^t
  dbar.t.perm<-t(dbar.perm)
  # calculate t^2 value, in matrix notation: n*dbar^t * S^-1 * dbar
  t2.perm<-n.perm*(dbar.t.perm%*%S.inv.perm%*%dbar.perm)
  # transform T^2 to F ratio
  F.perm <- ((n.perm-k.perm)/((n.perm-1)*k.perm))*t2.perm
  sig[i]<-ifelse (F.perm>=F,1,0)
  F.permb[i]<-F.perm
}

# compute p-value by summing sig and dividing by the number of permutations
# be sure and add 1 to account for the "observed" F ratio, it has been seen at least once!!! (why
pvalues equaling 0 make no sense.
p.perm<-(sum(sig)+1)/(nperm+1);p.perm
hist(c(F.permb,F));abline(v=F,col="red",lwd=2)

# Now let's get a p value by permutin without ANY assumptions regarding T, or F distributions,
much, much more powerful, and much, much simpler
# the question becomes: "can we get a greater DISTANCE between the mean-vectors by random
chance alone? "

# number of permutations
nperm<-9999
sig<-numeric(nperm)
m1 <- apply(group1, 2, mean);m1
m2 <- apply(group2, 2, mean);m2
```

```
D<-dist(rbind(m1,m2))
D.permb<-numeric(nperm)

for (i in 1:nperm)  {
cat("Perm No.", i,"\n")
  # permute sample
  g1.perm<-gmatrix[sample(nrow(gmatrix),nrow(group1)),]
  g2.perm<-gmatrix[sample(nrow(gmatrix),nrow(group2)),]
  m1.perm <- apply(g1.perm, 2, mean);m1
  m2.perm <- apply(g2.perm, 2, mean);m2
  D.perm<-dist(as.matrix(rbind(m1.perm,m2.perm)))
  D.permb[i]<-D.perm
  sig[i]<-ifelse (D.perm>=D,1,0)
}

# compute p-value by summing sig and dividing by the number of permutations
# be sure and add 1 to account for the "observed" D, as with F, it has been seen at least once!!! (why
pvalues equaling 0 make no sense.
p.perm<-(sum(sig)+1)/(nperm+1);p.perm

# plot the distribution of permuted vector distances against the "observed" D.
hist(c(D.permb,D));abline(v=D,col="red",lwd=2)
```