0921 code

# R code for performing TMA analysis

# Read in data

tma = read.table("tma505.dat",header=T)

# Subset the first three columns

tmamatrix = tma[,1:3]

# Goodness of fit statistics

TMA = as.matrix(tmamatrix)

# Compute mean and variance

muhat = apply(TMA,2,mean)

sigmahat = var(TMA)

d2 = NULL

for (i in 1:50) {

d2 = c(d2,(TMA[i,] - muhat)%\*%solve(sigmahat)%\*% (TMA[i,]-muhat))

}

plot(qchisq((1:50)/50,3),sort(d2),

xlab="Chi-squared percentiles, df = 3",ylab="d2")

# Plot 45 degree line

abline(0,1)

# One-sample Z statistic for testing mu = mu0 = (0.25,0.25,0.25)

n = dim(tmamatrix)[1]

mu0 = rep(0.25,3)

# Statistic

Z = n\*(muhat - mu0) %\*% solve(sigmahat) %\*%

(muhat-mu0)

# p-value

pval = 1 - pchisq(Z,df=3)

# One-sample T statistic for testing mu = mu0

# dimension of vector

p = 3

# Here, I pooled all constants into one statistic

T2 = (n\*(n-p)/(p\*(n-1)))\*(muhat - mu0)%\*% solve(sigmahat) %\*%

(muhat-mu0)

# p-value

pval = 1 - pf(T2,df1=p,df2=n-p)

# R function for implementing confidence intervals

# using simultaneous univariate intervals ideas

#

# est is the estimates of the parameter

#

# Note that the length of est must be p

confreg <- function(X,alpha=0.05){

n = dim(X)[1]

p = dim(X)[2]

s=cov(X)

simucr=matrix(0,p,2)

dg2=n-p

cr=qf((1-alpha),p,n-p)

cr1=sqrt(p\*(n-1)\*cr/(n-p))

se=sqrt(diag(s))/sqrt(n)

est = colMeans(X)

simucr[,1]=est-cr1\*se

simucr[,2]=est+cr1\*se

print("C.R. based on T^2")

print(simucr)

indvcr=matrix(0,p,2)

q=1-(alpha/2)

cr=qt(q,(n-1))

indvcr[,1]=est-cr\*se

indvcr[,2]=est+cr\*se

print("CR based on individual t")

print(indvcr)

bonfcr=matrix(0,p,2)

q=1-(alpha/(2\*p))

cr=qt(q,(n-1))

bonfcr[,1]=est-cr\*se

bonfcr[,2]=est+cr\*se

print("CR based on Bonferroni")

print(bonfcr)

asymcr=matrix(0,p,2)

cr=sqrt(qchisq((1-alpha),p))

asymcr[,1]=est-cr\*se

asymcr[,2]=est+cr\*se

print("Asymp. simu. CR")

print(asymcr)

return(list(simucr=simucr,indvcr=indvcr,bonfcr=bonfcr,asymcr=asymcr))}