HW#3

Haichuan Du

6/16/2020

```
library(snpar)
library(jmuOutlier) #For doing permutation tests.
```

##2

```
perm.test8452=function(x,y){
print(wilcox.test(x,y))
print(ks.test(x,y))
print(ansari.test(x,y))
print(paste("Sample1:"))
print(x)
print(paste("Sample2:"))
print(y)
}
```

```
set.seed(23)
x=rnorm(10)
y=rnorm(10)
perm.test8452(x,y)
```

```
##
##  Wilcoxon rank sum test
##
## data:  x and y
## W = 79, p-value = 0.02881
## alternative hypothesis: true location shift is not equal to 0
##
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  x and y
## D = 0.5, p-value = 0.1678
## alternative hypothesis: two-sided
##
##
##  Ansari-Bradley test
##
## data:  x and y
## AB = 58, p-value = 0.7102
## alternative hypothesis: true ratio of scales is not equal to 1
##
## [1] "Sample1:"
##  [1]  0.19321233 -0.43468211  0.91326710  1.79338809  0.99660511
##  [6]  1.10749049 -0.27808628  1.01920549  0.04543718  1.57577959
## [1] "Sample2:"
##  [1]  0.2182885 -1.0465353 -0.2886886  0.4815503 -1.2163764  0.3081369
##  [7] -0.5201783 -0.4423138 -0.5993128  1.2945778
```

##4

```
#Ansari-Bradley  with a location shift.
n=20 #Common sample size.
nruns=1000 #Number of runs to do.
nrej1=0 #For Ansari-Bradley test.

for (i in 0:4) {
  set.seed(0)
  nrej1=0
for (run in 1:nruns){
  samp1=rnorm(n, 0, 1)
  samp2=rnorm(n, 0, 1) + i
  if (ansari.test(samp1,samp2)$p.value<0.05){nrej1=nrej1+1}}
print(paste("Estimated alpha for delta = ", i , "Ansari-Bradley is:", nrej1/nruns))}
```

```
## [1] "Estimated alpha for delta =  0 Ansari-Bradley is: 0.038"
## [1] "Estimated alpha for delta =  1 Ansari-Bradley is: 0.018"
## [1] "Estimated alpha for delta =  2 Ansari-Bradley is: 0.001"
## [1] "Estimated alpha for delta =  3 Ansari-Bradley is: 0"
## [1] "Estimated alpha for delta =  4 Ansari-Bradley is: 0"
```

Comment: As the delat get bigger, the appha get smaller. The Ansari-Bradley is invalid for different location from normal distributions, but it is valid and robust for equal centers from normal distributions.