Stat 425 HW4 Solution

Fritz Scholz

1. The purpose of this homework is to understand the power behavior of the two-sample Wilcoxon test when sampling from normal populations which may differ from each other by a shift parameter Δ, i.e., $\mathcal{N}(\mu,\sigma^2)$ and $\mathcal{N}(\mu+\Delta,\sigma^2)$, respectively.
In particular, we want to compare the power function of the rank-sum test against that of the two-sample t-test. We also want to understand to what extent the asymptotic relative efficiency (ARE) $e_{W,t} = 3/\pi$ is reflected for finite sample sizes m and n. We want to use both normal approximations for the power function and explore their quality in relation to m and n.
This exercise offers opportunity for extra credit (to make up for previous losses) by extending the breadth of your investigation (other α, m and n). Provide your function codes, plots and a narrative that explains coherently what you have learned.
First note that the ranks of samples

$$X_1,\ldots,X_m \sim \mathcal{N}(\mu,\sigma^2) \qquad \text{and} \qquad Y_1,\ldots,Y_n \sim \mathcal{N}(\mu+\Delta,\sigma^2)$$

are the same as the ranks of the transformed samples

$$X_i' = (X_i-\mu)/\sigma \sim \mathcal{N}(0,1),\ i=1,\ldots,m \quad \text{and} \quad Y_j' = (Y_j-\mu)/\sigma \sim \mathcal{N}(\Delta/\sigma,1) = \mathcal{N}(\Delta',1),\ j=1,\ldots,n$$

since the common transformation $(\cdot-\mu)/\sigma$ does not alter the joint order relationships among X's and Y's. Hence the distribution of the rank-sum is the same, whether we sample from $\mathcal{N}(\mu,\sigma^2)$ and $\mathcal{N}(\mu+\Delta,\sigma^2)$ or from $\mathcal{N}(0,1)$ and $\mathcal{N}(\Delta',1)$ with $\Delta' = \Delta/\sigma$. Thus the power of the rank-sum test does not depend on μ and it depends on Δ and σ only through the ratio $\Delta' = \Delta/\sigma$. A corresponding property holds for the two-sample t-test, namely its power depends on μ, Δ and σ only through $\Delta' = \Delta/\sigma$. Note however, that in both cases (Wilcoxon and t-test) the sample sizes m and n affect the power.

Write a function `Ranksum.sim=function(m=10,n=10,alpha=.05,Nsim=10000,Delta.p=.5){...}` that simulates the distribution of the Wilcoxon rank-sum statistic W_s for samples of sizes m and n from $\mathcal{N}(0,1)$ and $\mathcal{N}(\Delta',1)$, respectively ($\Delta' \equiv$ `Delta.p`). By distribution is meant a vector `Ws.vec` of length `Nsim`, containing the results from calculating the rank-sums W_s for `Nsim` simulations of independent samples of sizes `m` and `n` from $\mathcal{N}(0,1)$ and $\mathcal{N}(\Delta',1)$, respectively. Run these simulations in a loop (`for(i in 1:Nsim){...}`) with appropriate initialization of `Ws.vec` (remember HW3).
We consider one-sided rank-sum tests which reject $H_0:\Delta=0$ whenever $W_s \ge c_\alpha$, where c_α is the lowest integer value such that $P_{H_0}(W_s \ge c_\alpha) \le \alpha$. To find the appropriate c_α you may use `qwilcox` but understand that `qwilcox(p,m,n)` returns the smallest L such that $P_{H_0}(W_{XY} \le L) \ge p$ and realize the appropriate relationship between W_{XY} and W_s. Explain your reasoning in coming up with c_α.
`Ranksum.sim` should produce a named vector[1] with components representing

$$\texttt{Nsim},\ \texttt{m},\ \texttt{n},\ \alpha,\ c_\alpha,\ \alpha_c,\ \Delta',\ P_{\Delta'}(W_s \ge c_\alpha)$$

where $\alpha_c = P_0(W_s \ge c_\alpha)$ is the achieved significance level ($\le \alpha$) when using $c = c_\alpha$ as critical point. $P_{\Delta'}(W_s \ge c_\alpha)$ represents the power of the test at the alternative Δ', the quantity of main interest to us. While building this function use `Nsim` $=$ `100` for faster debugging.

[1] For example, you name a vector `out` $=$ `c(x,y,z)` via `names(out)` $=$ `c("x.name","name.y","z")`.

As a check run Ranksum.sim for $\texttt{Nsim} = 10000$ and $\Delta' = 0$. Your power should then be close to the achieved significance level α_c, which of course depends on m and n though qwilcox.
Next, write a function

$$\texttt{power.fun} = \texttt{function}(\texttt{Nsim} = 10000, \texttt{alpha} = .05, \texttt{m} = 10, \texttt{n} = 10, \texttt{fac} = 3/\texttt{pi})\{\ldots\}$$

that evaluates Ranksum.sim for Delta.p in $\texttt{Delta.vec} = \texttt{seq}(0,2,\texttt{length.out} = 21)$ and then plots $P_{\Delta'}(W_s \geq c_\alpha)$ against Delta.p over the grid vector Delta.vec. In a loop store the calculated values of $P_{\Delta'}(W_s \geq c_\alpha)$ in a vector power.vec of same length as Delta.vec. Superimposed on this plot

```
plot(Delta.vec,power,type="l",xlab=expression(Delta*minute==Delta/sigma),
            ylab=expression(Pi(Delta*minute)==Pi(Delta/sigma)),ylim=c(0,1))
```

add the power function of the two-sample t-test, evaluated over the same grid. Do this by using the $\texttt{lines}(\texttt{x},\texttt{y})$ command for appropriate vectors x and y. The power function values for the t-test can be obtained in vectorized mode (since we use the vector argument Delta.vec) via

$$\texttt{power.t} = 1 - \texttt{pt}(\texttt{qt}(1 - \texttt{alpha.c}, \texttt{m}+\texttt{n}-2, 0), \texttt{m}+\texttt{n}-2, \texttt{Delta.vec}/\texttt{sqrt}(1/\texttt{m}+1/\texttt{n}))$$

Explain this last command in terms of the fact that the distribution of the two-sample t-statistics is a noncentral t-distribution with $m+n-2$ degrees of freedom and noncentrality parameter

$$\delta = \frac{\Delta'}{\sqrt{1/m+1/n}} = \frac{\Delta}{\sigma\sqrt{1/m+1/n}}.$$

We expect the power of the t-test to be slightly higher than the power of the Wilcoxon rank-sum test. To get a better match of the power functions recompute the power of the t-test when m and n are reduced by the factor $\texttt{fac} = 3/\texttt{pi} = 3/\pi$, which represents the ARE of the Wilcoxon test relative to the t-test. This adjustment (only for the power of the t-test) is possible since pt and qt allow non-integer degrees of freedom. However, non-integer sample sizes don't make sense in the application of the t-test. What can you say about the quality of the match-up? Note that you can make both comparisons by using $\texttt{fac} = 1$ and $\texttt{fac} = 3/\texttt{pi}$ in the argument sequence to power.fun.
In spite of the quality of the match-up what aspect makes the rank-sum test preferable? Does the above match-up of power suggest a way to plan the sample sizes for the rank-sum test when dealing with normal shift alternatives (without simulating the W_s distribution for Δ)?
Now add to this plot the power as computed by the two normal approximations and add a legend in the upper left corner using the legend(...) command, e.g.,

```
legend(0,1,c("simulated power of Ws",
      paste("non-central t power (fac =",round(fac,3),")"),
     "power: normal approx. 1","power: normal approx. 2"),
      col=c("black","blue","red","orange"),lty=1:4,bty="n")
```

Make sure the various lines(...) commands use the appropriate colors and lty parameters. Also add the following annotation to your plot.

```
text(max(Delta.vec),0,substitute(N[sim]==xNsim~", "~m ==xm~",  "
  ~n ==xn~",  "~alpha==xalpha~",  "~alpha[c]==xalpha.c,
  list(xNsim=Nsim,xm=m,xn=n,xalpha=alpha,xalpha.c=round(alpha.c,3))),adj=1)
```

Show these plots for $m = 10$ and $n = 10$ and $\alpha = .05$. Discuss your results.

Note that I have given you two instances of writing mathematical expressions (Greek) in your plot via `expression` and `substitute`. For more on this see the link to "An Approach to Providing Mathematical Annotation in Plots" by Paul Murrell and Ross Ihaka that I provided on the class web page.

Optional (no need to do all): While the plots should look fine for $m = n = 10$ they could use some scaling improvement for $m = n = 5$ or $m = n = 30$. Try to implement this in an automatic fashion by using the second normal approximation to find an appropriate U (corresponding to approximate power .99) to get an adjustable grid vector `Delta.vec=seq(0,U,length.out=21)`. What about two-sided tests? What would you have to change if you were to compare the power of t-test and Wilcoxon test for non-normal shift alternatives. Note that the non-central t-distribution no longer applies.

The code for the two functions `Ranksum.sim` and `power.fun` is given at the end. Note that lines 3 and 4 of `power.fun` set up the plotting grid from 0 to the .99-quantile based on the second normal approximation. This adapts the relevant plotting range to the choice of m and n.
We checked `Ranksum.sim` by running it for `Delta.p=0`

```
> Ranksum.sim(10,10,.05,100000,0)
        Nsim            m            n        alpha      c.alpha      alpha.c
1.000000e+05 1.000000e+01 1.000000e+01 5.000000e-02 1.280000e+02 4.460478e-02
     Delta.p    power.sim
0.000000e+00 4.644000e-02
```

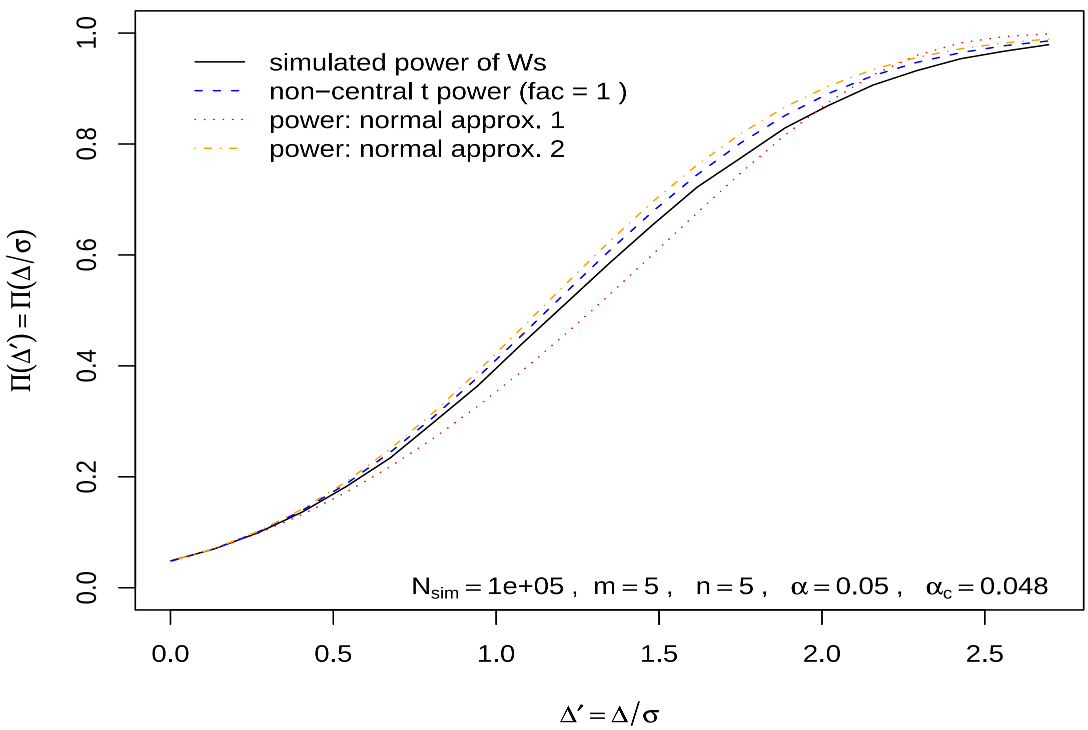
The values `4.460478e-02` and `4.644000e-02` are reasonably close to each other.
As for finding c_α using `qwilcox` we give the following explanation. `qwilcox(1-alpha,m,n)` gives us the smallest $k = k_\alpha$ such that $P_0(W_{XY} \leq k) \geq 1-\alpha$, i.e., the smallest k such that $P_0(W_{XY} \geq k+1) = \alpha_c \leq \alpha$ and (since $W_s = W_{XY} + n(n+1)/2$) thus the smallest k such that $P_0(W_s \geq k+1+n(n+1)/2) \leq \alpha$. Thus our critical point should be

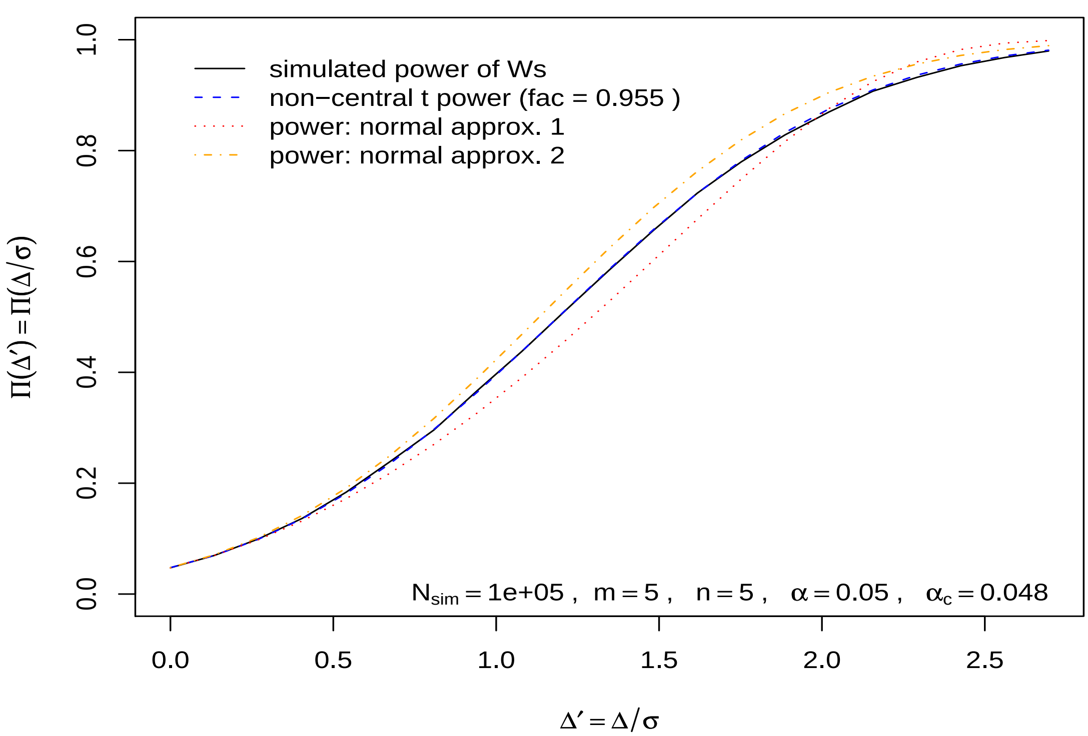
$$c_\alpha = k+1+n(n+1)/2 = \texttt{qwilcox}(1-\texttt{alpha},\texttt{m},\texttt{n})+1+\texttt{n}(\texttt{n}+1)/2$$

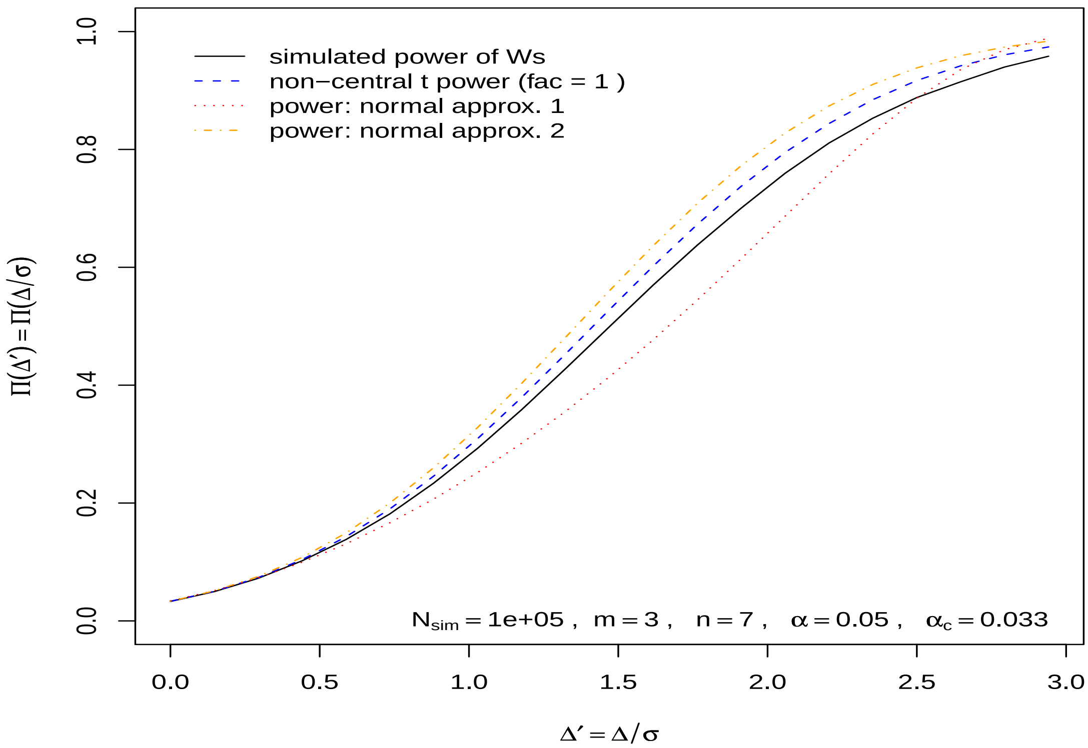
and is implemented in line 4 of `Ranksum.sim`.
The line calculating the power of the t-test

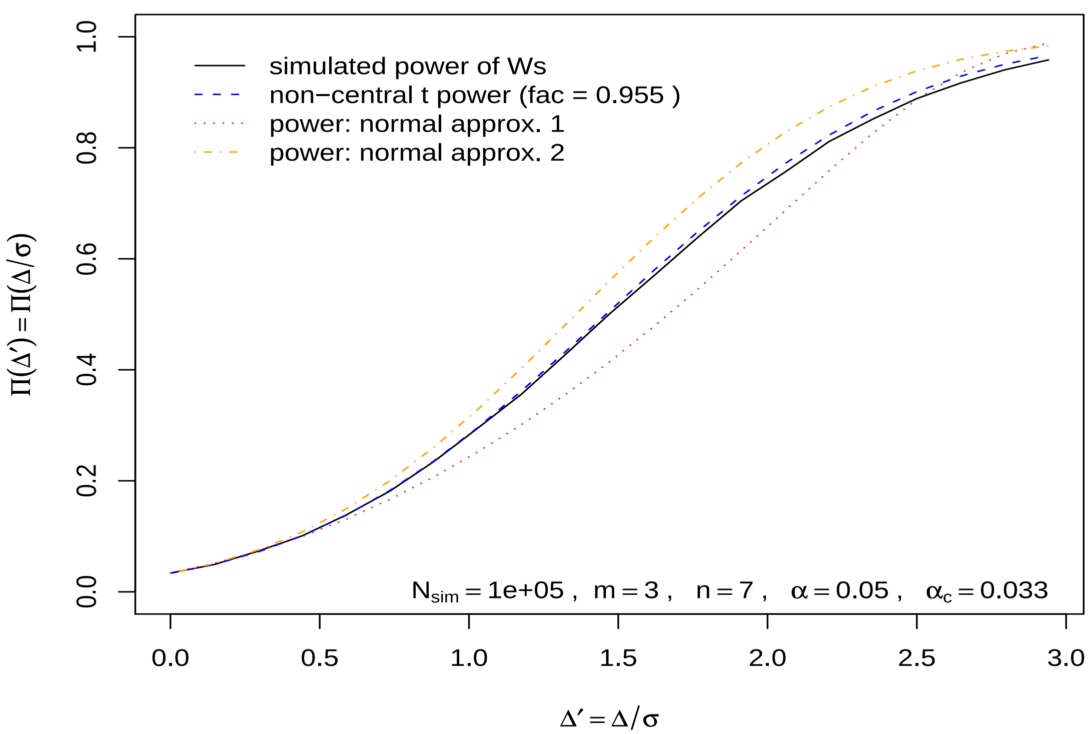
$$\texttt{power.t} = 1-\texttt{pt}(\texttt{qt}(1-\texttt{alpha.c},\texttt{m}+\texttt{n}-2,0),\texttt{m}+\texttt{n}-2,\texttt{Delta.vec}/\texttt{sqrt}(1/\texttt{m}+1/\texttt{n}))$$

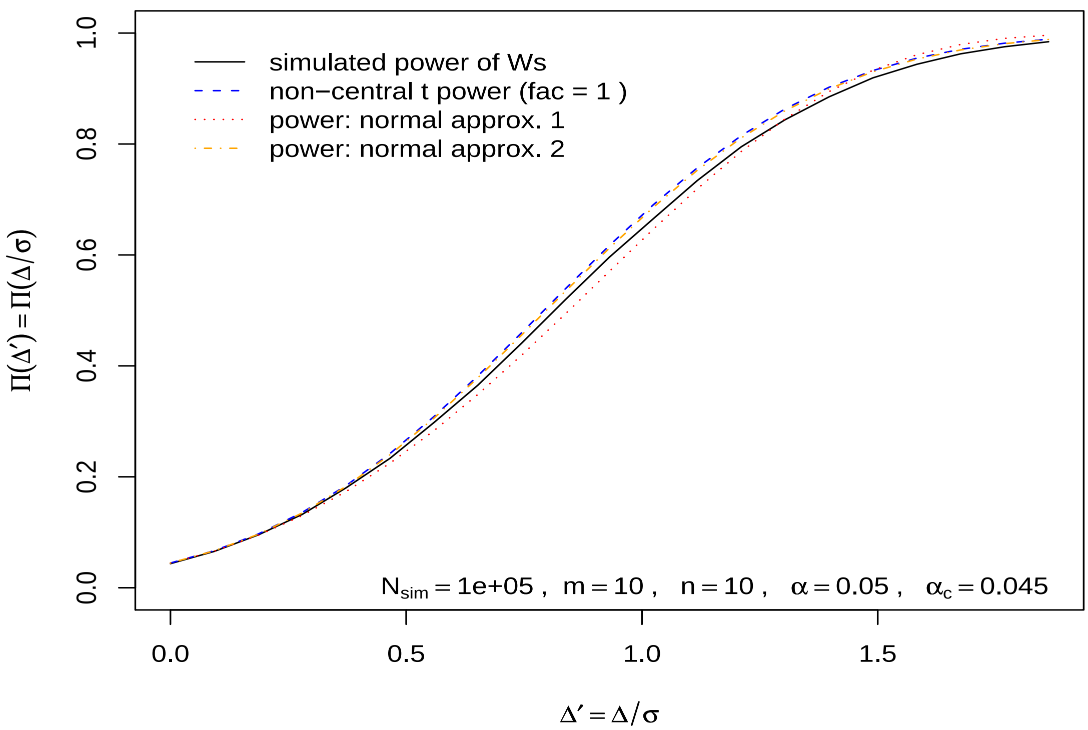
is explained as follows. `1-pt(x,m+n-2,delta)` gives the probability that the two-sample t-statistic with `m+n-2` degrees of freedom is $\geq x$, i.e., rejects the hypothesis when `delta` is the noncentrality parameter. For sampled normal distribution $\mathcal{N}(\mu,\sigma^2)$ and $\mathcal{N}(\mu+\Delta,\sigma^2)$ this noncentrality parameter is $\texttt{delta} = \Delta/(\sigma\sqrt{1/m+1/n})$. For the critical point $x = c_\alpha = \texttt{qt}(1-\alpha.\texttt{c},\texttt{m}+\texttt{n}-2,0)$ this rejection probability will be α when $H_0: \Delta = 0$ is true, i.e., make it a level α test. For any other `delta` it will give you the power of that t-test.
The following plots show the results for various runs of `power.fun` as indicated by the annotations.

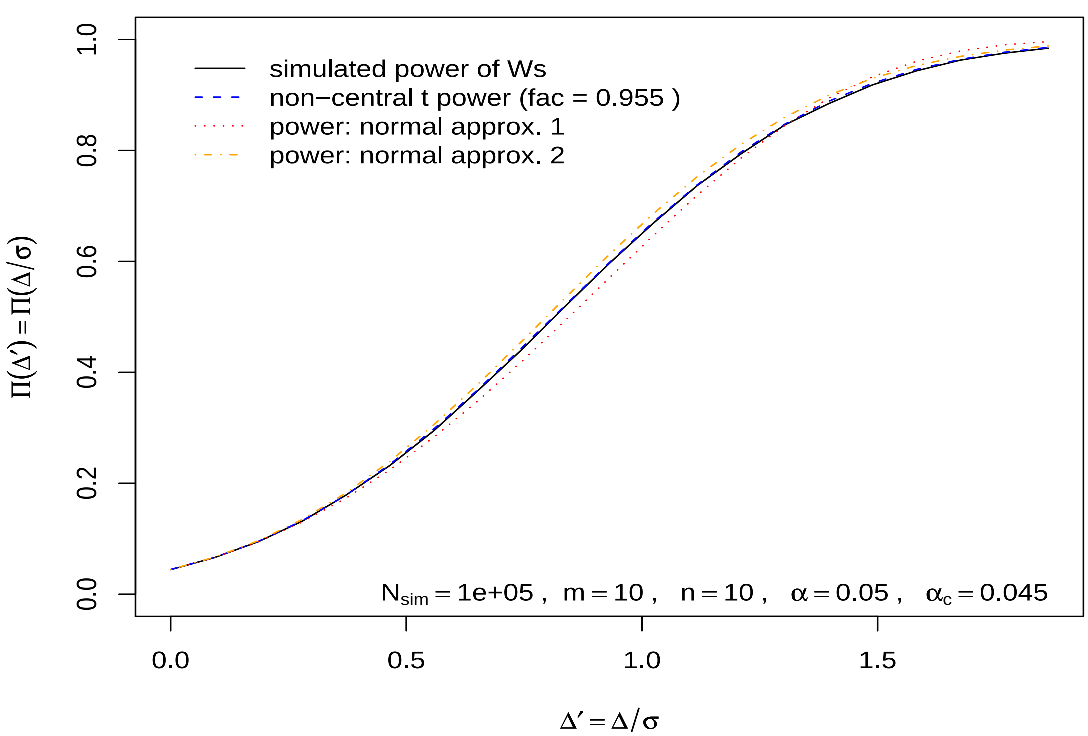

simulated power of Ws
non−central t power (fac = 1)
power: normal approx. 1
power: normal approx. 2
$N_{sim} = 1e+05$, $m = 5$, $n = 5$, $\alpha = 0.05$, $\alpha_c = 0.048$
$\Pi(\Delta') = \Pi(\Delta/\sigma)$
$\Delta' = \Delta/\sigma$
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
2.5

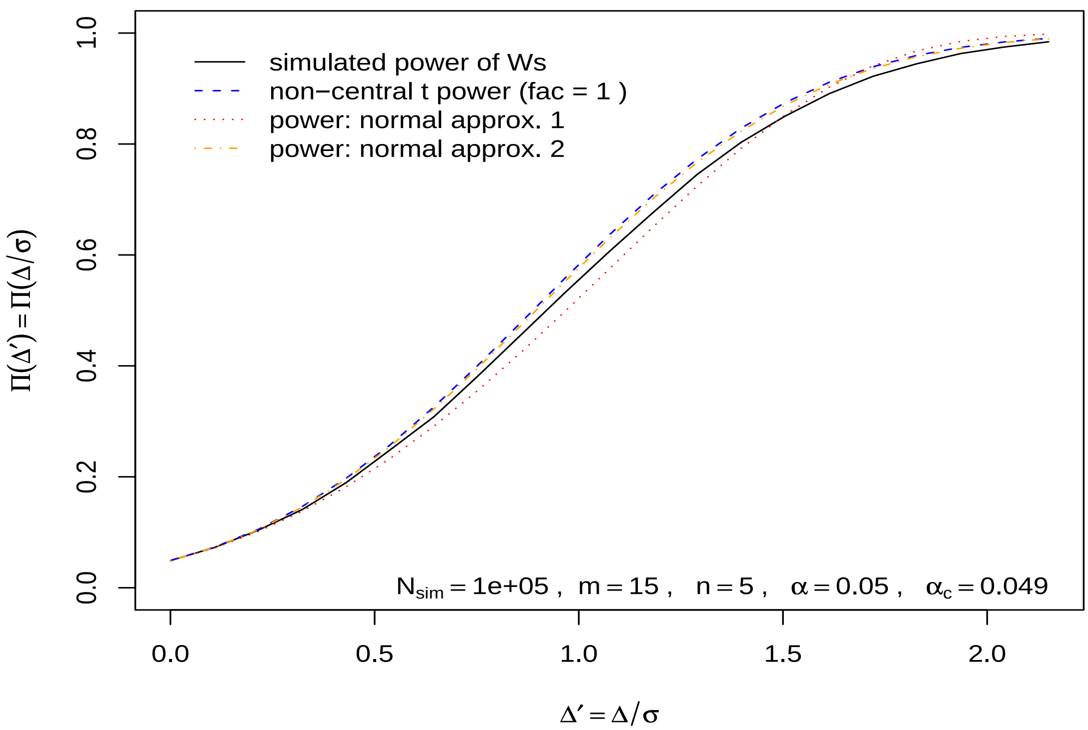

simulated power of Ws
non−central t power (fac = 0.955)
power: normal approx. 1
power: normal approx. 2
$N_{sim} = 1e+05$, $m = 5$, $n = 5$, $\alpha = 0.05$, $\alpha_c = 0.048$
$\Pi(\Delta') = \Pi(\Delta/\sigma)$
$\Delta' = \Delta/\sigma$
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
2.5


simulated power of Ws
non−central t power (fac = 1)
power: normal approx. 1
power: normal approx. 2
$N_{sim} = 1e+05$, $m = 3$, $n = 7$, $\alpha = 0.05$, $\alpha_c = 0.033$
$\Pi(\Delta') = \Pi(\Delta/\sigma)$
$\Delta' = \Delta/\sigma$

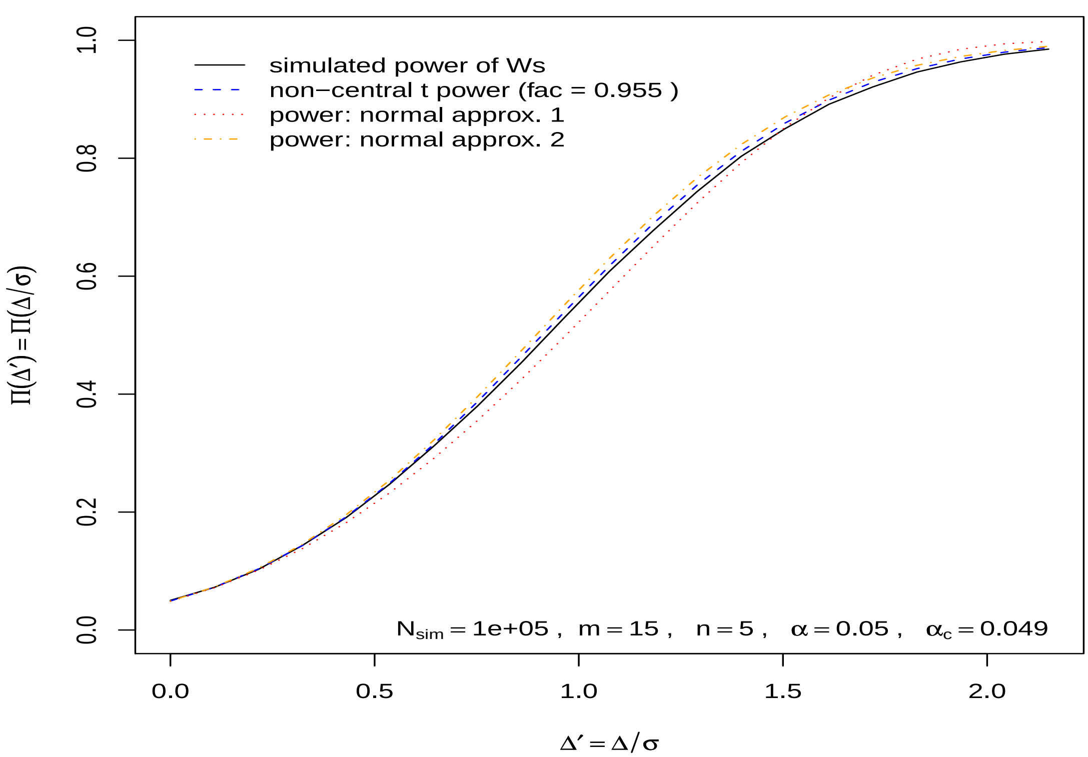

simulated power of Ws
non−central t power (fac = 0.955)
power: normal approx. 1
power: normal approx. 2
$N_{sim} = 1e+05$, $m = 3$, $n = 7$, $\alpha = 0.05$, $\alpha_c = 0.033$
$\Pi(\Delta') = \Pi(\Delta/\sigma)$
$\Delta' = \Delta/\sigma$


simulated power of Ws
non−central t power (fac = 1)
power: normal approx. 1
power: normal approx. 2
$N_{sim} = 1e+05$, $m = 10$, $n = 10$, $\alpha = 0.05$, $\alpha_c = 0.045$
$\Pi(\Delta') = \Pi(\Delta/\sigma)$
$\Delta' = \Delta/\sigma$
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5

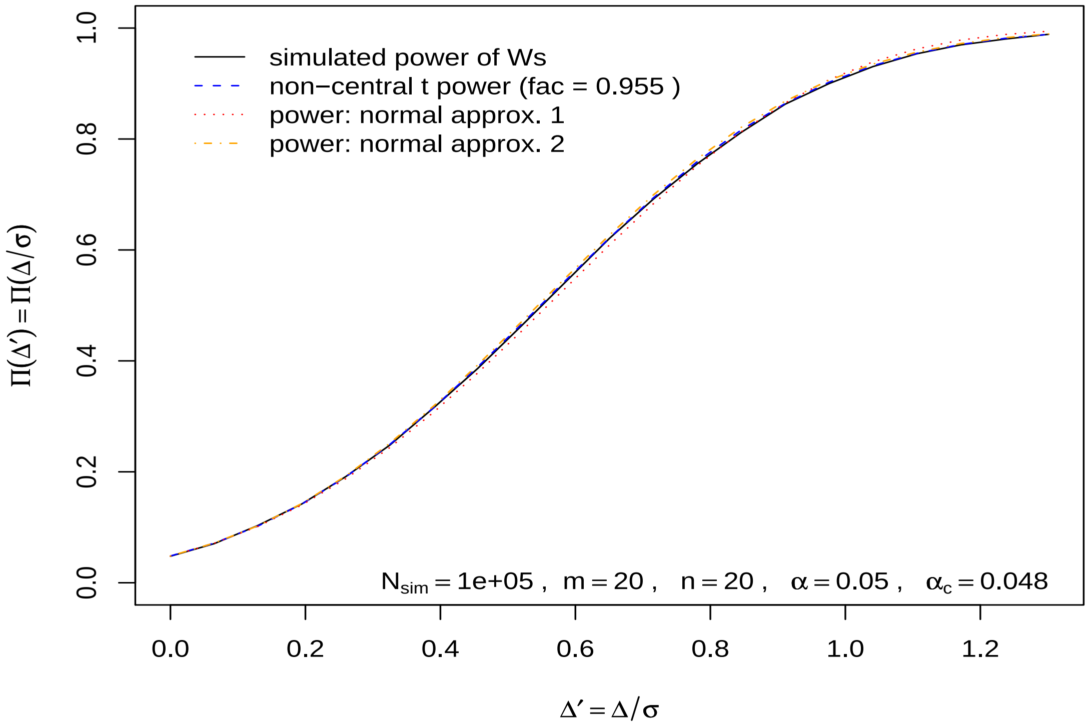

simulated power of Ws
non−central t power (fac = 0.955)
power: normal approx. 1
power: normal approx. 2
$N_{sim} = 1e+05$, $m = 10$, $n = 10$, $\alpha = 0.05$, $\alpha_c = 0.045$
$\Pi(\Delta') = \Pi(\Delta/\sigma)$
$\Delta' = \Delta/\sigma$
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5

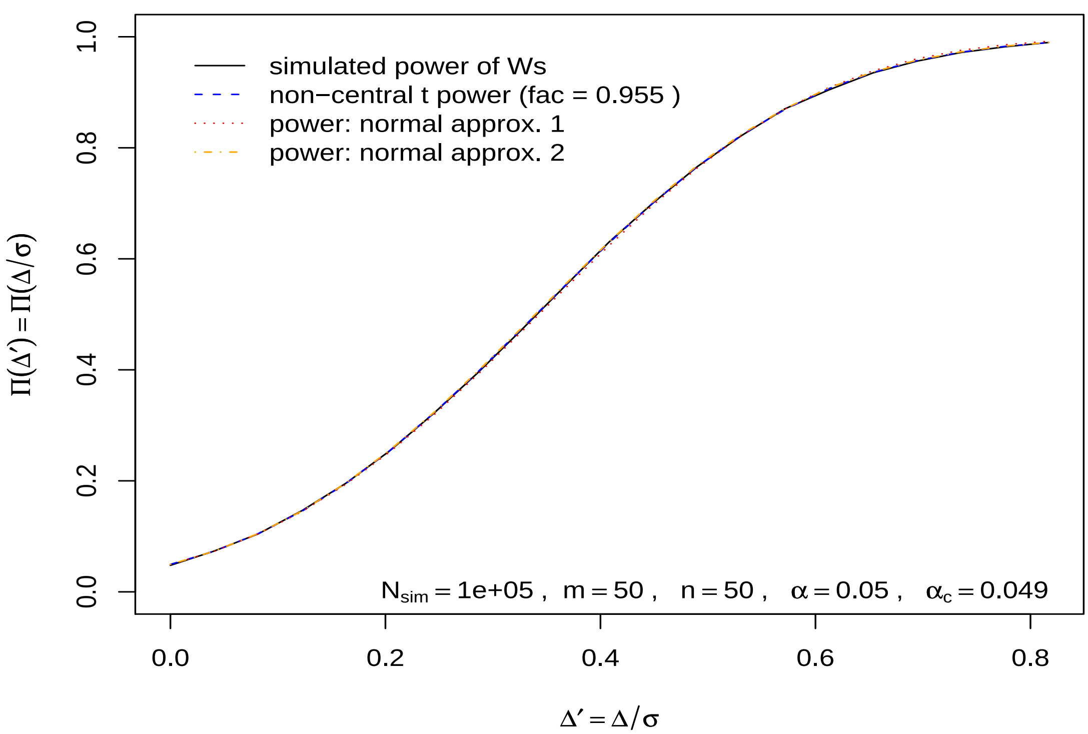


simulated power of Ws
non−central t power (fac = 1)
power: normal approx. 1
power: normal approx. 2
$N_{sim} = 1e+05$, $m = 15$, $n = 5$, $\alpha = 0.05$, $\alpha_c = 0.049$
$\Pi(\Delta') = \Pi(\Delta/\sigma)$
$\Delta' = \Delta/\sigma$
0.0
0.2
0.4
0.6
0.8
1.0
0.5
1.5
2.0



simulated power of Ws
non−central t power (fac = 0.955)
power: normal approx. 1
power: normal approx. 2
$N_{sim} = 1e+05$, $m = 15$, $n = 5$, $\alpha = 0.05$, $\alpha_c = 0.049$
$\Pi(\Delta') = \Pi(\Delta/\sigma)$
$\Delta' = \Delta/\sigma$
0.0
0.2
0.4
0.6
0.8
1.0
0.5
1.5
2.0


simulated power of Ws
non−central t power (fac = 0.955)
power: normal approx. 1
power: normal approx. 2
$N_{sim} = 1e+05$, $m = 20$, $n = 20$, $\alpha = 0.05$, $\alpha_c = 0.048$
$\Pi(\Delta') = \Pi(\Delta/\sigma)$
$\Delta' = \Delta/\sigma$
0.0
0.2
0.4
0.6
0.8
1.0
1.2


simulated power of Ws
non−central t power (fac = 0.955)
power: normal approx. 1
power: normal approx. 2
$N_{sim} = 1e+05$, $m = 50$, $n = 50$, $\alpha = 0.05$, $\alpha_c = 0.049$
$\Pi(\Delta') = \Pi(\Delta/\sigma)$
$\Delta' = \Delta/\sigma$
0.0
0.2
0.4
0.6
0.8

Some observations: When comparing the Wilcoxon test with the t-test for the same respective sample sizes (using `fac=1`) the power of the former is clearly lower than the power of the latter, as expected. This is illustrated for sample size combinations $m = n = 5$,$m = 3, n = 7$, $m = n = 10$, and $m = 15, n = 5$. When we adjust (reduce) the sample sizes in the t-test power function by changing m and n to $m' = m \cdot 3/\pi$ and $n' = n \cdot 3/\pi$ as suggested by the ARE (i.e., asymptotically the t-test should require only $3/\pi$ of the Wilcoxon test sample sizes for equal power) we find a very good match between the adjusted power of the t-test and the simulated power (`Nsim=100000`) of the Wilcoxon test. This matchup is especially good when $m = n$, as seen for $m = n = 5$ and $m = n = 10$. Even for the unbalanced sample sizes $m = 3$, $n = 7$ and $m = 15$, $n = 5$ this match is still very good (especially when comparing it with the normal approximations), namely very good for power $\leq .5$ and with slight separation between the power curves for power $> .5$. The match is better for the higher total sample size $N = 15+5$ than for $N = 3+7$, as expected. Also shown are the corresponding plots for $m = n = 20$ and $m = n = 50$ with the ARE-adjusted t-test power function. The latter and the normal approximations agree fairly well for such larger sample sizes.
The normal approximations are quite poor for low $N = m+n = 10$, especially for $N = 3+7$, although they are in the right ballpark. The normal approximation improved for $N = m+n = 20$, but that improvement is again somewhat negatively impacted for the unbalanced case $m = 15, N = 5$. Generally the ARE-adjusted t-test power curves are better than either of the normal approximations for small to moderate sample sizes $m = n$.
Because of the good approximation quality of the ARE-adjusted t-test power and because the calculation of that power is essentially instantaneous it would make sense to use the t-test power to plan sample sizes $m = n$ for the Wilcoxon test. For any $m = n$ find the appropriate c_α and α_c by using `qwilcox` and `pwilcox` and then use that α_c to evaluate the t-test power when using $m' = n' = (3/\pi)m = (3/\pi)n$ for the $\Delta' = \Delta/\sigma$ of interest. Based on our previous plots this power should be very close to the power of the Wilcoxon test at the same shift alternative and for the same significance level. This t-test power calculation is instantaneous and does not require time consuming simulation, which would slow matters considerably when trying to iterate.
In spite of the good match-up of power and the resulting lower required sample size for the t-test it is preferable to use the Wilcoxon test, because its significance level is correct under $H_0 : \Delta = 0$, no matter what the underlying distribution F in the shift model is. The t-test would only approximately be distribution-free in that regard, provided the variance of F is finite.
The code for a two-sided test version of `Ranksum.sim` (namely `Ranksum2.sim`) is attached at the end. By proper modification (as commented inside the function) it can be used to generate samples from an exponential shift model. By running that for $m \neq n$ (sufficiently different) one could illustrate the non-symmetric behavior for the $\Pi(\Delta) \neq \Pi(-\Delta)$ in the exponential shift model. The modified version is denoted by `Ranksum2exp.sim` (not shown here) and it produced the following confirmation of the above asymmetry.

```
> Ranksum2exp.sim(20,5,.05,10000,1.5,"both")
        Nsim            m            n        alpha      k.alpha     alpha.c
1.000000e+04 2.000000e+01 5.000000e+00 5.000000e-02 8.000000e+01 4.234896e-02
     Delta.p    power.sim
1.500000e+00 9.175000e-01
> Ranksum2exp.sim(20,5,.05,10000,-1.5,"both")
         Nsim             m             n         alpha       k.alpha
 1.000000e+04  2.000000e+01  5.000000e+00  5.000000e-02  8.000000e+01
```

```
     alpha.c       Delta.p     power.sim
 4.234896e-02 -1.500000e+00  8.336000e-01
```

The difference between the powers `9.175000e-01` and `8.336000e-01` is substantial, i.e., confirms the asymmetry of Π at $\Delta = \pm 1.5$ when $m = 20$ and $n = 5$.

```
Ranksum.sim=function(m=10,n=10,alpha=.05,Nsim=10000,Delta.p=.5){
k.alpha=qwilcox(1-alpha,m,n)
alpha.c=1-pwilcox(k.alpha,m,n)
c.alpha=k.alpha+1+n*(n+1)/2
Ws.vec=rep(0,Nsim)
for(i in 1:Nsim){
   x=rnorm(m); y=rnorm(n)+Delta.p
   z=c(x,y)
   Ws.vec[i]=sum(rank(z)[m+(1:n)])
}
power.sim=mean(Ws.vec>=c.alpha)
xout=c(Nsim,m,n,alpha,c.alpha,alpha.c,Delta.p,power.sim)
names(xout)=c("Nsim","m","n","alpha","c.alpha",
    "alpha.c","Delta.p","power.sim")
xout
}
```

and

```
power.fun=function(Nsim=10000,alpha=.05,m=10,n=10,fac=3/pi,PDF=F){
if(fac==1){fac0=1}else{fac0=round(1000*fac,0)}
alph=round(alpha*100,0)
if(PDF==T) pdf(file=paste("RankSumPowerm",m,"n",n,"fac",
          fac0,"alpha",alph,".pdf",sep=""),width=7)
U=(qnorm(.99)+qnorm(1-alpha))*sqrt((m+n+1)*pi/(3*m*n))
Delta.vec=seq(0,U,length.out=21)
M=length(Delta.vec)
power=rep(0,M)
for(i in 1:M){
out=Ranksum.sim(m,n,alpha,Nsim,Delta.vec[i])
power[i]=out[8]
}
alpha.c=out[6]
k.alpha=out[5]
power.t=1-pt(qt(1-alpha.c,fac*(m+n)-2,0),
   fac*(m+n)-2,Delta.vec/(sqrt(1/m+1/n)*sqrt(1/fac)))
plot(Delta.vec,power,type="l",
     xlab=expression(Delta*minute==Delta/sigma),
     ylab=expression(Pi(Delta*minute)==Pi(Delta/sigma)),
     ylim=c(0,1))
lines(Delta.vec,power.t,col="blue",lty=2)
```

```
if(!exists("pmnorm"))library(mnormt)
p1=pnorm(Delta.vec/sqrt(2))
p2=rep(0,M)
for(i in 1:M){
p2[i]=pmnorm(c(Delta.vec[i]/sqrt(2),Delta.vec[i]/sqrt(2)),
      c(0,0),varcov=matrix(c(1,.5,.5,1),ncol=2))
}
p3=p2
meanWXY=m*n*p1
varWXY=m*n*p1*(1-p1)+m*n*(m+n-2)*(p2-p1^2)
power.n1=1-pnorm((k.alpha-n*(n+1)/2-.5-meanWXY)/sqrt(varWXY))
power.n2=pnorm(sqrt(3*m*n/((m+n+1)*pi))*Delta.vec-qnorm(1-alpha.c))
lines(Delta.vec,power.n1,col="red",lty=3)
lines(Delta.vec,power.n2,col="orange",lty=4)
legend(0,1,c("simulated power of Ws",
      paste("non-central t power (fac =",round(fac,3),")"),
     "power: normal approx. 1","power: normal approx. 2"),
      col=c("black","blue","red","orange"),lty=1:4,bty="n")
text(max(Delta.vec),0,substitute(N[sim]==xNsim~", "~m ==xm~",  "
  ~n ==xn~",  "~alpha==xalpha~",  "~alpha[c]==xalpha.c,
if(PDF==T) dev.off()
}

Ranksum2.sim=function(m=10,n=10,alpha=.05,Nsim=10000,Delta.p=0,alternative="higher"){
#==========================================================================================
# this function simulates the power of one- and two-sided Wilcoxon rank-sum tests,
# depending on the choice alternative = "higher", "lower" or "both".
# samples are drawn from normal and shifted normal distributions.
# Commented in is the change to make if you want to sample from an exponential
# and shifted exponential distribution. Modifications for other sampled shift model
# distributions should be obvious.
#==========================================================================================
if(alternative=="higher" | alternative=="lower"){
     k=qwilcox(1-alpha,m,n)+1
   }else{
     k=qwilcox(1-alpha/2,m,n)+1
}
Ws.vec=rep(0,Nsim)
for(i in 1:Nsim){
   x=rnorm(m); y=rnorm(n)+Delta.p
   # changing the previous line to
   # x=rexp(m); y=rexp(n)+Delta.p
   # would generate samples from an exponential
   # and a shifted exponential distribution.
   # The exponential distribution is not symmetric.
   # Thus the power of the two-sided test would not be
```

```
    # symmetric around the origin for m != n.
    z=c(x,y)
    Ws.vec[i]=sum(rank(z)[m+(1:n)])
}
if(alternative=="higher"){
    cWs=k+n*(n+1)/2
    power=mean(Ws.vec>=cWs)
}
if(alternative=="lower"){
    cWs=m*n-k+n*(n+1)/2
    power=mean(Ws.vec<=cWs)
}
if(alternative!="higher" & alternative != "lower"){
    cWs1=k+n*(n+1)/2
    cWs2=m*n-k+n*(n+1)/2
    power=mean(Ws.vec<=cWs2 | Ws.vec>=cWs1)
}
if(alternative=="higher" | alternative=="lower"){
        alpha.c=1-pwilcox(k-1,m,n)
    }else{
        alpha.c=2*(1-pwilcox(k-1,m,n))
}
xout=c(Nsim,m,n,alpha,k,alpha.c,Delta.p,power)
names(xout)=c("Nsim","m","n","alpha","k.alpha","alpha.c","Delta.p","power.sim")
xout
}
```