# Laboratorium: Lista 1
# Podstawy symulacji komputerowych

Statystyka w zastosowaniach

Sylwia Majchrowska
Matematyka

14 marca 2016

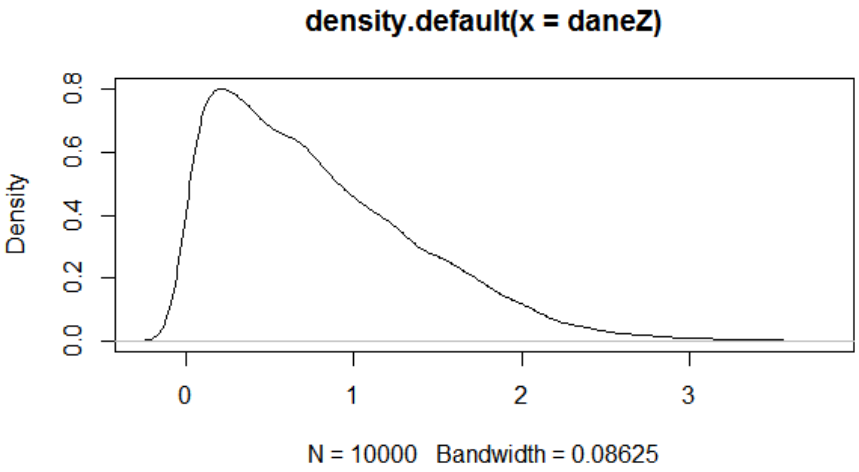
## Spis treści

# 1 Wstęp

Rozważany jest problem testowania dla dwóch prób: $X_1, ..., X_n \sim N(\mu_1, \sigma^2)$, $Y_1, ..., Y_n \sim N(\mu_2, \sigma^2)$, $H_0 : \mu_i = \mu_2$ vs $H_A : \mu_i \neq \mu_2$, za pomocą statystyki testowej $Z = \frac{|\overline{X} - \overline{Y}|}{\sigma}\sqrt{\frac{n}{2}}$, gdzie $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, $\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$.

# 2 Zadanie pierwsze

Należało wyznaczyć wartości krytyczne dla $Z$ na poziomach istotności $\alpha = 0.1$ oraz $\alpha = 0.05$. W tym celu dobrano odpowiednio liczbę prób ($n = 10\ 000$) oraz liczbę przeprowadzanych symulacji (także 10 000). Następnie wygenerowano dane $X_1, ..., X_n \sim N(0, 1)$ i $Y_1, ..., Y_n \sim N(0, 1)$ dla wszystkich 10 000 powtórzeń formując odpowiedni wektor $Z$, którego współrzędne były obliczonymi statystykami dla poszczególnych symulacji. Gęstość rozkładu statystyki testowej przedstawia wykres 1.




Rysunek 1: Rozkład statystyki Z.

Formalnie kwantyl rzędu $q$ $(0 < q < 1)$ jest taką liczbą $x_q$, że $q * 100\%$ elementów próby posiada wartość badanej cechy nie większą niż $x_q$.

Aby obliczyć odpowiedni kwantyl, wektor $Z$ posortowano, a następnie wskazano jego współrzędną odpowiadającą zaokrąglonemu w dół indeksowi $(1 - \alpha) \cdot \textit{liczba\_symulacji}$.

Przykładowe wyniki symulacji:

- dla $\alpha = 0.1$ wartość krytyczna dla $Z$ wyniosła 1.671101,
- dla $\alpha = 0.05$ wartość krytyczna dla $Z$ wyniosła 1.979808.

Wyniki są zgodne z oczekiwaniami.

Kod źródłowy zadania 1 w języku R:

```
liczba_prob = 10000
liczba_symulacji = 10000
daneZ = rep(0, liczba_symulacji)
for(i in 1: liczba_symulacji){
dane = rnorm(liczba_prob)
daneX = rnorm(liczba_prob)
daneY = rnorm(liczba_prob)
daneZ[i] = abs(mean(daneX)-mean(daneY))
          *sqrt(liczba_prob/2)
}
plot(density(daneZ))

kwantyl = function(alfa, dane){
dane1 = sort(dane)
return (dane1[floor((1-alfa)*length(dane))])
}

A0.1 = kwantyl(0.1, daneZ)
A0.05 = kwantyl(0.05, daneZ)
```

# 3 Zadanie drugie

Zadanie polegało na zaprojektowaniu badania dla estymacji prawdopodobieństwa popełnienia błędu pierwszego rodzaju $\alpha$. Należało ustalić taką liczbę powtórzeń $m$ eksperymentu, aby $P(|\hat{p} - p| \leqslant 0.1p) \simeq 0.95$.

Dla dużych $m$ rozkład $B(m,p) \sim N(mp, \sigma\sqrt{mp(1-p)})$, co wiemy z Centralnego Twierdzenia Granicznego.

Estymatorem prawdopodobieństwa popełnienia błędu pierwszego rodzaju jest $\hat{p} = \frac{x}{m} \sim N(p, \sqrt{p(1-p)/m})$, gdzie $x-$ liczba przypadków, w których popełniamy błąd pierwszego rodzaju.

$$P(|\hat{p} - p| \leqslant 0.1p) = P\left(\frac{|\hat{p} - p|}{\sqrt{p(1-p)}}\sqrt{m} \leqslant \frac{0.1p}{\sqrt{p(1-p)}}\sqrt{m}\right)$$

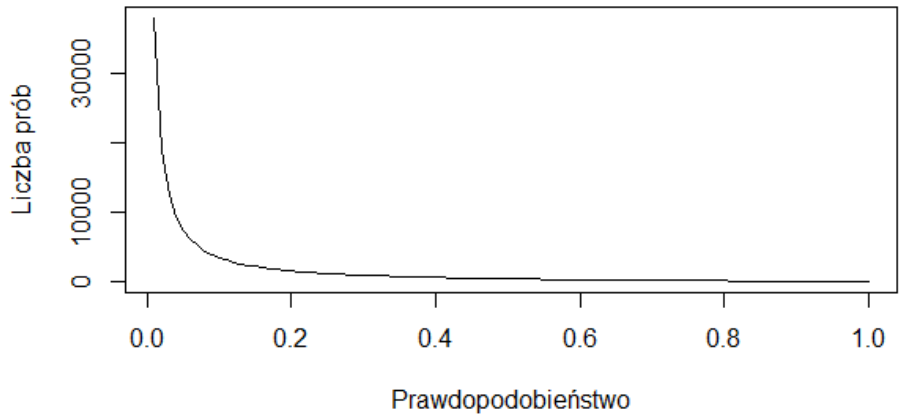
$$P(Z \leqslant 0.1\sqrt{\frac{p}{1-p}}\sqrt{m}) = 0.95$$

Co oznacza, że

$$0.1\sqrt{\frac{p}{1-p}}\sqrt{m} \approx 1.96$$

Ostatecznie

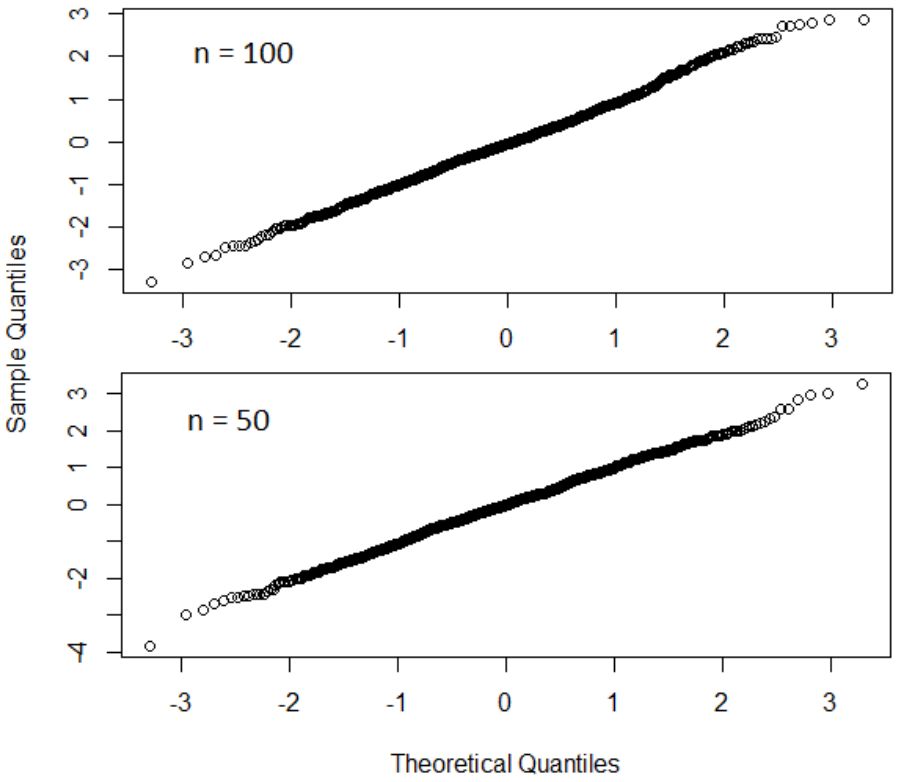
$$m \approx 19.6^2 \cdot \frac{1-p}{p}$$




Rysunek 2: Funkcja zależności potrzebnej ilości powtórzeń eksperymentu od prawdopodobieństwa popełnienia błędu pierwszego rodzaju.

Aby przeprowadzić eksperyment weryfikujący powyższe obliczenia napisano funkcję $moneta(prob)$, generującą wektor zer i jedynek (dla jedynek losowanych z prawdopodobieństwem $prob$ [w powyższych rozważaniach oznaczone jako $p$], gdzie odpowiadają one eksperymentom, w których popełniono błąd I rodzaju). W kolejnym kroku przeprowadzono 1000 symulacji dla prób, których liczność wyznaczono posługując się funkcją $number(prob)$. Zwrócony przez nią wynik zaokrąglono w górę. Następnie utworzono wektor prawdopodobieństw $temp$ [w powyższych rozważaniach oznaczone jako $\hat{p}$] wylosowania jedynek, dla każdej z 1000 symulacji. Ostatnim krokiem było wyznaczenie wielkości $|\hat{p} - p| - 0.1p$ (jej wartości dla poszczególnych symulacji przypisano współrzędnym wektora $temp$). Na samym końcu należało sprawdzić, jaka część współrzędnych w ten sposób utworzonego wektora jest mniejsza bądź równa 0. To wszystko zostało zrealizowane za pomocą funkcji $test(prob)$. Zwracane przez nią wartości oscylują w okolicach 0.95, co jest zgodne z przewidywaniami.

W przypadku, gdy obserwacje $X_i \sim Exp(\lambda)$ i $Y_i \sim Exp(\lambda) + \mu$, gdzie $\lambda = 1$, możemy przybliżać dany rozkład rozkładem normalnym nawet dla prób o małej liczności. Eksperyment został przeprowadzony dla $n = 50$ oraz $n = 100$. Uzyskany wynik przedstawiono na wykresach kwantylowo-kwantylowych (wykres 3), na których widać oczekiwane liniowe zależności. O dążeniu do rozkładu normalnego dla wygenerowanych danych orzeka również test Kolmogorova-Smirnova.




Rysunek 3: Wykresy kwantylowo-kwantylowe dla statystyki testowej $Z$ obliczonej dla prób pochodzących z rozkładów wykładniczych.

Kod źródłowy zadania 2 w języku R:

```
#a)
number <- function(prob){
  return (((19.6)^2)*(1-prob)/prob)
}
number(0.1)
pvalue=seq(0.01,1, by=0.01)
numberTossing=number(pvalue)
plot(pvalue,numberTossing,   type="l", lwd=1,
         lty=1,col=c("black"),
         ylab = "Liczba prob", xlab="Prawdopodobienstwo")

#b)

moneta <- function(prob, len){
  return (sample(c(1,0), len, replace=TRUE,
         prob=c(prob,1-prob)))
}
```

```
test <- function(prob){ #eksperyment weryfikujacy
  repNum=trunc(number(prob)+1)
  SymNum=1000
   mactest=matrix(moneta(prob,repNum*SymNum), ncol=SymNum)
   temp=apply(mactest,2,mean)
   temp=abs(temp-prob)-0.1*prob
   return (length(temp[temp<=0])/length(temp))
}

#c)

n=100
symNum=1000
Xe=matrix(rexp(n*symNum,1), ncol=symNum)
Ye=matrix(rexp(n*symNum,1), ncol=symNum)
Xme=apply(Xe,2,mean)
Yme=apply(Ye,2,mean)
Ze=(Xme-Yme)*sqrt(n/2)
qqnorm(Ze)
#hist(Ze)
ks.test(Ze,"pnorm")
```

# 4 Zadanie trzecie

Zadaniem było wyznaczenie mocy testu, czyli prawdopodobieństwa odrzucenia $H_0$ w funkcji $\delta = \frac{|\mu_1 - \mu_2|}{\sigma}$.

Moc testu w zależności od poziomu istotności $\alpha$ oznaczamy jako

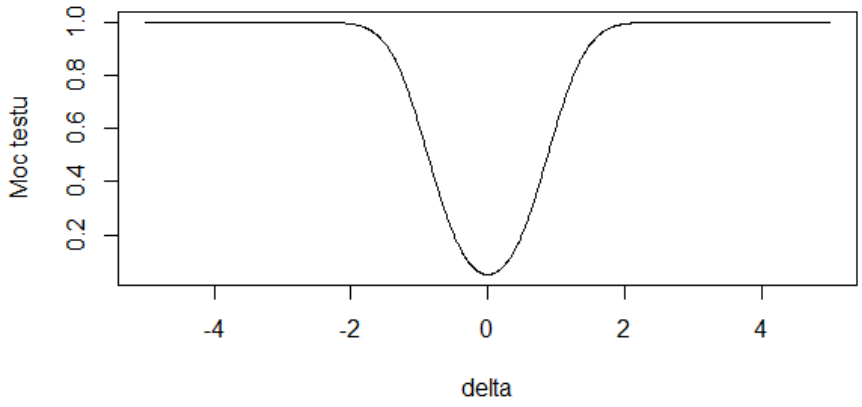
$$P(H_0 \text{ odrzucono}) = P(|X| > X_{(1-\alpha/2)}) = P(X > X_{(1-\alpha/2)}) + P(X < X_{(\alpha/2)})$$

Dodając do rozważań parametr niecałkowalności $\delta$ otrzymujemy równanie

$$P(X-\delta > X_{(1-\alpha/2)}-\delta)+P(X-\delta < X_{(\alpha/2)}-\delta) = 1-\Phi(X_{(1-\alpha/2)}-\delta)+\Phi(X_{(\alpha/2)}-\delta)$$

W przypadku, gdy $\alpha = 0.05$ otrzymamy

$$P\left(H_0 \text{ odrzucono}\right) = 1 - \Phi\left(\Phi^{-1}\left(0.975\right) - \sqrt{\frac{n}{2}}\delta\right) + \Phi\left(\Phi^{-1}\left(0.025\right) - \sqrt{\frac{n}{2}}\delta\right)$$




Rysunek 4: Moc testu jako funkcja parametru $\delta$ dla $\alpha = 0.05$ oraz 10 obserwacji.

Moc testu rośnie wraz ze wzrostem parametru niecałkowalności $\delta$. Wraz ze wzrostem poziomu istotności $\alpha$ moc testu szybciej zbiega do 1. Moc testu uzależniona jest także od liczby obserwacji. Im dobrana próba będzie liczniejsza tym test posiadał będzie większą moc, czyli mniejsza będzie szansa popełnienia błędu.

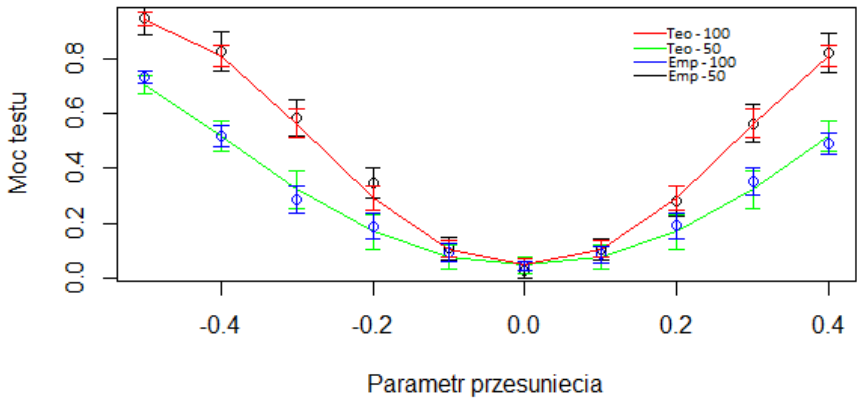
Kod źródłowy zadania 3 w języku R:

```
Moc <- function(a, delta, proba){
  return((1-pnorm(qnorm(1-a/2)-sqrt(proba/2)*delta))
         +pnorm(qnorm(a/2)-sqrt(proba/2)*delta))
}

X=seq(-5,5,by=0.01)
Y=Moc(0.05,X,10)

plot(X,Y,   type="l", lwd=1, lty=1,col=c("black"),
     xlab = "delta", ylab="Moc testu")
```

# 5 Zadanie czwarte

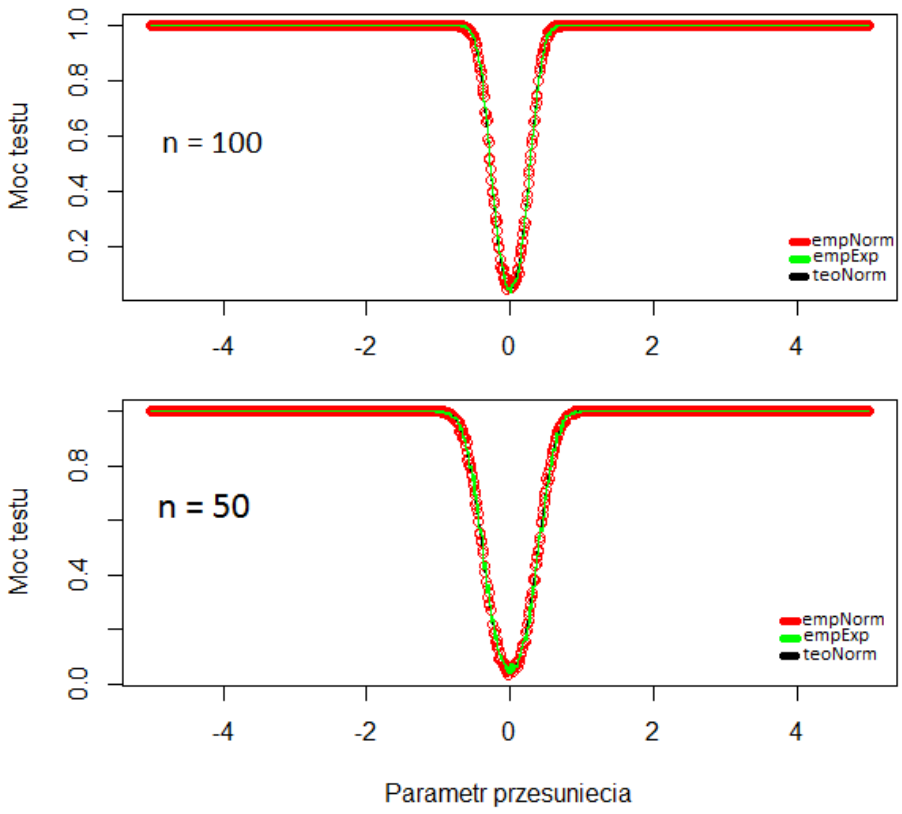
Początkowo estymacja funkcji mocy została dokonana dla 10 wartości parametru przesunięcia przy $n = 50$ oraz $n = 100$. Estymatory wyznaczono w oparciu o 500 powtórzeń eksperymentu. Na wykresie 5 przedstawiono wartości estymatora (punkty czarne i niebieskie) funkcji mocy oraz jej wartość teoretyczną (linia zielona i czerwona) dla obserwacji generowanych z rozkładu normalnego o dwóch różnych licznościach prób. Zaobserwowane charakterystyki estymatorów, jak i wartości wyznaczone teoretycznie, są do siebie zbliżone.

Odchylenie standardowe dla wyznaczonej wartości funkcji mocy wyniesie $\sqrt{\text{Var}\hat{p}} = \sqrt{\frac{p(1-p)}{m}}$.




Rysunek 5: Moc testu jako funkcja parametru przesunięcia dla obserwacji generowanych z rozkładu normalnego (wartości empiryczne - oznaczone kolorem czarnym dla $n = 50$ i niebieskim dla $n = 100$ - oraz teoretyczne - oznaczone kolorem zielonym dla $n = 50$ i czerwonym dla $n = 100$) wraz z wyliczonymi wartościami odchylenia standardowego.

Następnie estymacja funkcji mocy została dokonana dla 1001 wartości parametru przesunięcia przy $n = 50$ oraz $n = 100$. Estymatory wyznaczono w oparciu o 500 powtórzeń eksperymentu. Na wykresie 6 przedstawiono wartości estymatora (linia czerwona) funkcji mocy oraz jej wartość teoretyczną (linia czarna) dla obserwacji generowanych zarówno z rozkładu normalnego, jak i wykładniczego, dla których funkcję mocy wyznaczono empirycznie (linia zielona). Zaobserwowane charakterystyki pokrywają się.




Rysunek 6: Moc testu jako funkcja parametru przesunięcia dla obserwacji generowanych z rozkładu normalnego (wartość empiryczna - kolor czerwony - i teoretyczna - linia czarna) oraz wykładniczego - linia zielona.

Kod źródłowy zadania 4 w języku R:

```
#a)
critical <- function(x){ #przedzial krytyczny
  wynik = 0
  if(x<(-1.96) || x>1.96){
    wynik=1}
  return(wynik)
}

Moc <- function(a,delta,proba){ #moc testu teoretycznie
  return((1-pnorm(qnorm(1-a/2)-sqrt(proba/2)*delta))
         +pnorm(qnorm(a/2)-sqrt(proba/2)*delta))
}

MocNormil <- function(delta){ #moc testu empirycznie
  n = 100 #liczba prob
  symNum = 500 #liczba symulacji
  Xe=matrix(rnorm(n*symNum),ncol=symNum)
  Ye=matrix(rnorm(n*symNum)+delta,ncol=symNum)
  Xme=apply(Xe,2,mean)
  Yme=apply(Ye,2,mean)
  Ze=abs(Xme-Yme)*sqrt(n/2) #statystyka testowa
  return (mean(apply(as.matrix(Ze),1,critical)))
}

MocNorm <- function(delta){ #moc testu empirycznie
  n = 50 #liczba prob
  symNum = 500 #liczba symulacji
  Xe=matrix(rnorm(n*symNum),ncol=symNum)
  Ye=matrix(rnorm(n*symNum)+delta,ncol=symNum)
  Xme=apply(Xe,2,mean)
  Yme=apply(Ye,2,mean)
  Ze=abs(Xme-Yme)*sqrt(n/2) #statystyka testowa
  return (mean(apply(as.matrix(Ze),1,critical)))
}

wek = seq(-0.5,0.4,by=0.1)
Z=Moc(0.05,wek,100)
bladteo2 = sqrt(Z*(1-Z)/100)

Zw=Moc(0.05,wek,50)
bladteo = sqrt(Z*(1-Z)/50)

spr2 = apply(as.matrix(wek),1,MocNormil)#n = 100
blademp2 = sqrt(spr2*(1-spr2)/100)

spr= apply(as.matrix(wek),1,MocNorm)#n = 50
blademp = sqrt(spr*(1-spr)/50)
```

```
plot(wek,spr2, type="p", col="black", ylab = "Moc testu",
     xlab="Parametr przesuniecia")
lines(wek,Zw,col="green",type="l")
arrows(wek, Zw-bladteo, wek, Zw+bladteo,
       length=0.05, col="green", angle=90, code=3)
arrows(wek, spr2-blademp, wek, spr2+blademp,
       length=0.05, angle=90, code=3)
lines(wek,spr, type="p", col="blue")
lines(wek,Z,col="red",type="l")

arrows(wek, Z-bladteo2, wek, Z+bladteo2,
       length=0.05, col="red", angle=90, code=3)
arrows(wek, spr-blademp2, wek, spr+blademp2,
       length=0.05, col="blue", angle=90, code=3)

#b) Przyklad dla n = 50

X=seq(-5,5,by=0.01)
Y=apply(as.matrix(X),1,MocNorm)

MocEmp <- function(shift){ #rozklad wykladniczy
  n=50
  symNum=500
  Xe=matrix(rexp(n*symNum,1),ncol=symNum)
  Ye=matrix(rexp(n*symNum,1),ncol=symNum)
  Xme=apply(Xe,2,mean)
  Yme=apply(Ye,2,mean)
  Ze=(Xme-Yme+rep(shift,symNum))*sqrt(n/2)
  return (mean(apply(as.matrix(Ze),1,critical)))
}

W=apply(as.matrix(X),1,MocEmp)

Z=Moc(0.05,X,50)

plot(X, Y, col="red", ylab = "Moc testu",
     xlab="Parametr przesuniecia")
lines(X, Z, col="black",type="l")
points(X, W, type="s", col="green")
```