Testowanie Hipotez

Agnieszka Tracz

2024-01-06

Temat projektu:

Niech $X_1, \ldots, X_n$ będzie próbą prostą z rozkładu $N(m, \sigma^2)$. Rozważmy 3 testy o rozmiarach $\alpha = 0.05$ hipotezy $H_0 : m = 0$ przeciwko alternatywie $H_1 : m \neq 0$ - test studenta, test Wilcoxona oraz test Znaków. Narysować na jednym rysunku wykresy mocy empirycznych dla $n = 50$, $\sigma = 1$ w zależności od m na przedziale $[-2, 2]$. Wykonać jeszcze 2 kolejne rysunki mocy empirycznych dla $\sigma = 3$ i $\sigma = 6$.

Test studenta:

Testujemy: $H_0 : m = 0$ przeciwko alternatywie $H_1 : m \neq 0$

P- wartość dla testu Tstudenta:

```
tstudenttest <- function(m, n, sigma ) {

  X <- rnorm(n, m, sigma)
  T <- sqrt(1/n) * (sum(X - 0)) / sqrt(sum((X - mean(X))^2) / (n - 1))
  P <- 2 * (1 - pt(abs(T), df = n - 1))

  return(P)
}

tstudenttest(0,50,1)
```

```
## [1] 0.6388471
```

Moc empiryczna w teście Tstudenta:

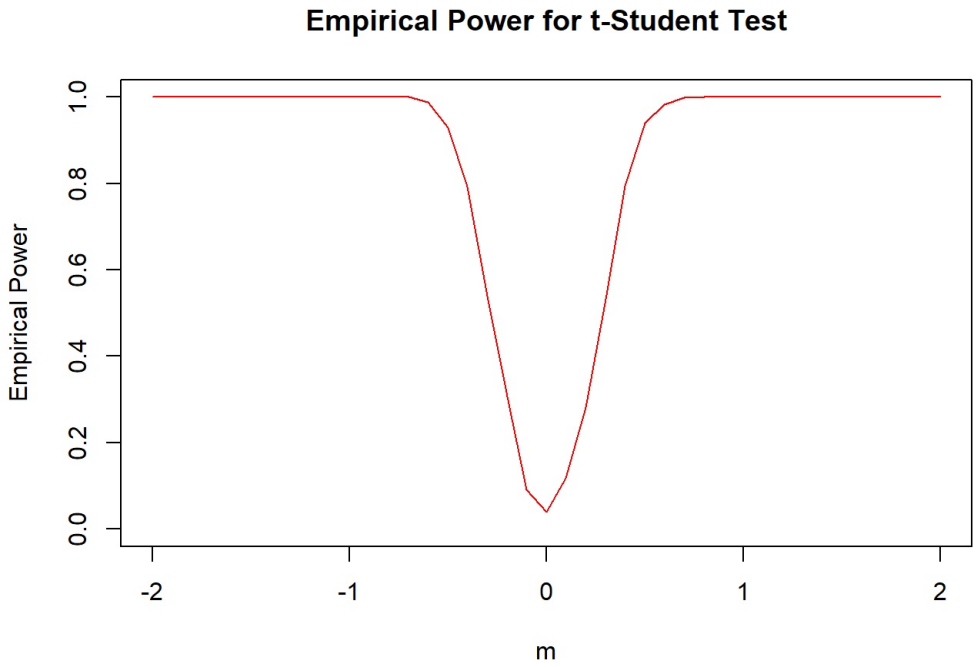
```
power_tstudent <- function(m, n, sigma) {
  num_simulations <- 1000  # liczba symulacji
  power <- 0

  for (i in 1:num_simulations) {

    X <- rnorm(n, m, sigma)
    T <- sqrt(1/n) * (sum(X - 0)) / sqrt(sum((X - mean(X))^2) /     (n - 1))
    P <- 2 * (1 - pt(abs(T), df = n - 1))
    if(P<0.05) power <- power+1
  }
  power <- power/num_simulations
  return(power)
}
```

Wykresy mocy empirycznych dla n = 50, σ = 1,3,6 w zależności od m na przedziale [−2 , 2]

```
m_values <- seq(-2, 2, by = 0.1)
n <- 50
```

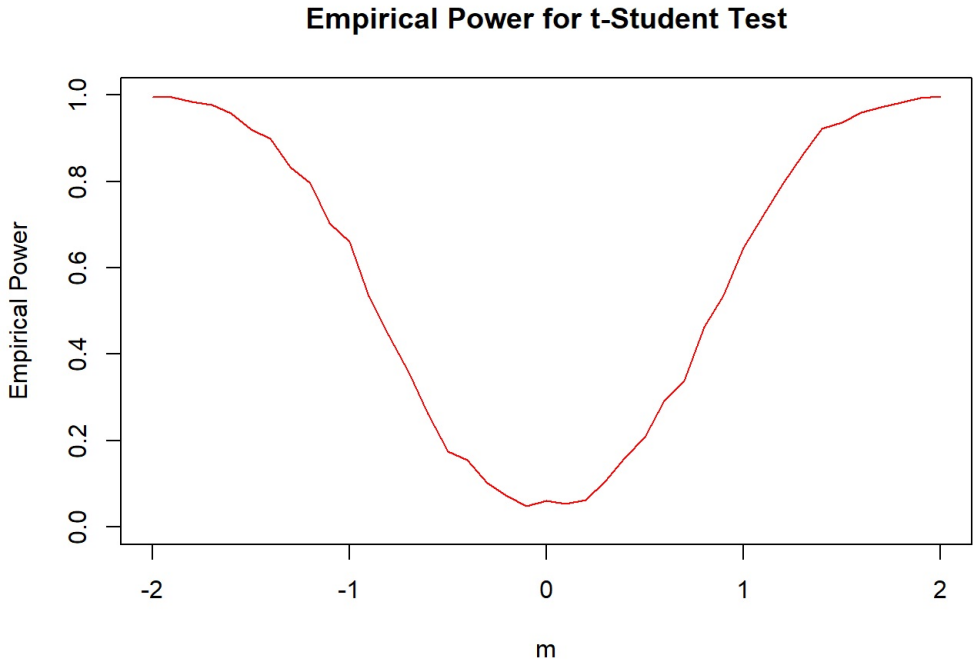
σ = 1

```
sigma <- 1
power_values <- sapply(m_values, function(m) power_tstudent(m, n, sigma))
plot(m_values, power_values, type = "l", col = "red",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power for t-Student Test",
     xlim = c(-2, 2), ylim = c(0, 1))
```



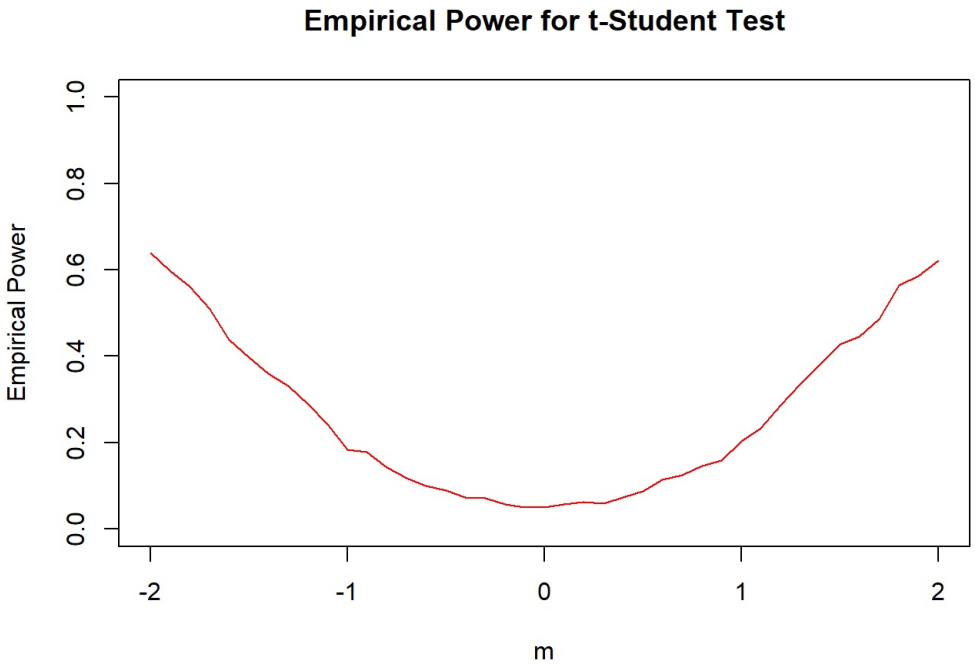
$\sigma = 3$

```
sigma <- 3
power_values <- sapply(m_values, function(m) power_tstudent(m, n, sigma))
plot(m_values, power_values, type = "l", col = "red",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power for t-Student Test",
     xlim = c(-2, 2), ylim = c(0, 1))
```



$\sigma = 6$

```
sigma <- 6
power_values <- sapply(m_values, function(m) power_tstudent(m, n, sigma))
plot(m_values, power_values, type = "l", col = "red",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power for t-Student Test",
     xlim = c(-2, 2), ylim = c(0, 1))
```



Test Wilcoxona

P- wartość dla testu Wilcoxona:

```
testWilcoxona <- function(m,n,sigma){

    X <- rnorm(n, m, sigma)
    R<-rank(abs(X),ties.method = "random")
    M <- 0
    for(i in 1:n)
    {
      if(X[i]>0){M=M+R[i]}
    }
    M=M/n
    W <- (M-(n+1)/4)/sqrt((n+1)*(2*n+1)/(24*n))
    P <- 2*(1-pnorm(abs(W),0,1))
    return(P)
}
testWilcoxona(0,50,1)
```

```
## [1] 0.965351
```

Dokładna P-wartość obliczona metodą MC:

```
testWilcoxonaMC <- function(m,n,sigma){
  X<- rnorm(n,m,sigma)
  R<-rank(abs(X),ties.method = "random")
  M <- 0
  for(i in 1:n)
  {
    if(X[i]>0){M=M+R[i]}
  }
  M=M/n
  W<-(M-(n+1)/4)/sqrt((n+1)*(2*n+1)/(24*n))
  WMC<-c()
  MC<-100000
  for (k in 1:MC)
  {
    ZMC <- runif(n,-1,1)
    RMC<-rank(abs(ZMC))
    MMC<-0
    for(i in 1:n)
    {
      if(ZMC[i]>0){MMC=MMC+RMC[i]}
    }
    MMC=MMC/n
    WMC[k]<-(MMC-(n+1)/4)/sqrt((n+1)*(2*n+1)/(24*n))
  }
  pMC<-0
  for (k in 1:MC)
  {
    if(abs(WMC[k])>=abs(W)){pMC=pMC+1}
  }
  pMC=pMC/MC
  pMC
  return(pMC)
}

testWilcoxonaMC(0,50,1)
```

```
## [1] 0.39204
```

Moc empiryczna w teście Wilcoxona:

```
power_Wilcoxon <- function(m, n, sigma) {
  num_simulations <- 1000  # liczba symulacji
  power <- 0
  for (i in 1:num_simulations) {
    X <- rnorm(n, m, sigma)
    R <- rank(abs(X), ties.method = "random")
    M <- sum(R * (X > 0)) / n
    W <- (M - (n + 1) / 4) / sqrt((n + 1) * (2 * n + 1) / (24 * n))
    p <- 2*(1-pnorm(abs(W)))
    if(p<0.05) power <- power+1
  }
  power <- power/num_simulations
  return(power)
}
power_Wilcoxon(0,50,1)
```

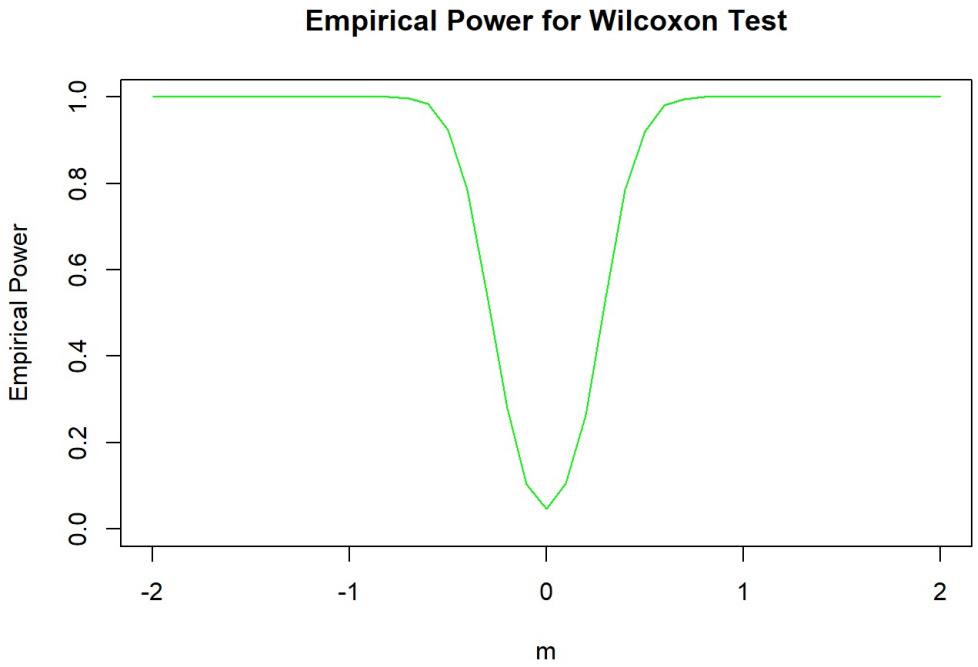
```
## [1] 0.044
```

Wykresy mocy empirycznych dla n = 50, σ = 1,3,6 w zależności od m na przedziale [−2 , 2]

```
m_values <- seq(-2, 2, by = 0.1)
n <- 50
```

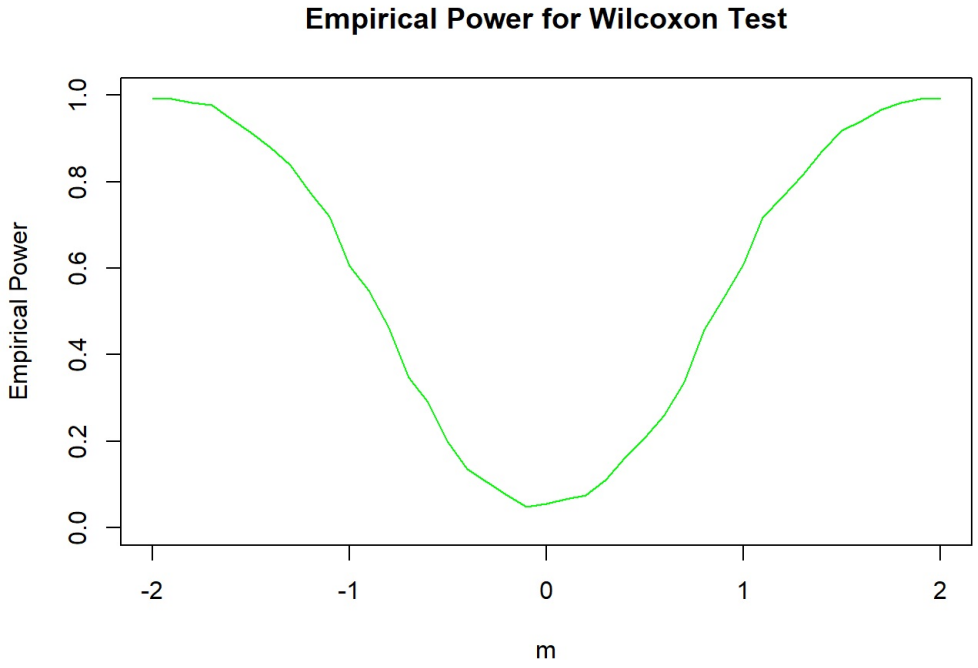
σ = 1

```
sigma <- 1
power_values_W <- sapply(m_values, function(m) power_Wilcoxon(m, n, sigma))

plot(m_values, power_values_W, type = "l", col = "green",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power for Wilcoxon Test",
     xlim = c(-2, 2), ylim = c(0, 1))
```



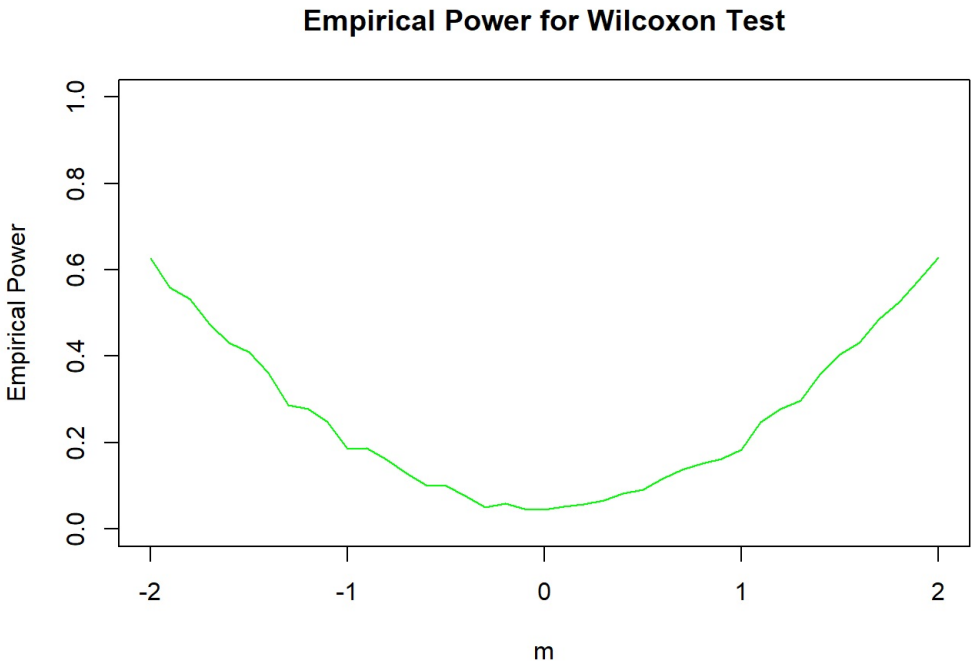
σ = 3

```
sigma <- 3
power_values_W <- sapply(m_values, function(m) power_Wilcoxon(m, n, sigma))

plot(m_values, power_values_W, type = "l", col = "green",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power for Wilcoxon Test",
     xlim = c(-2, 2), ylim = c(0, 1))
```



σ = 6

```
sigma <- 6
power_values_W <- sapply(m_values, function(m) power_Wilcoxon(m, n, sigma))

plot(m_values, power_values_W, type = "l", col = "green",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power for Wilcoxon Test",
     xlim = c(-2, 2), ylim = c(0, 1))
```



Test znaków

P- wartość dla testu Znaków:

```
testZnakow <- function(m,n,sigma,p){

  X <- rnorm(n, m, sigma)
  M <- 0
  for(i in 1:n)
  {
    if(X[i]>0){M=M+1}
  }
  M=M/n
  S<-sqrt(n)*(M-(1-p))/sqrt(p*(1-p))
  P <- 2*(1- pnorm(abs(S),0,1))
  return(P)
}
testZnakow(0,50,1,0.5)
```

```
## [1] 0.04771488
```

Dokładna P-wartość obliczona metodą MC:

```
testZnakowMC <- function(m,n,sigma,p){
  X <- rnorm(n,m,sigma)
  M <- 0
  for(i in 1:n)
  {
    if(X[i]>0){M=M+1}
  }
  M=M/n
  S<-sqrt(n)*(M-(1-p))/sqrt(p*(1-p))
  SMC<-c()
  MC<-1000
  for (k in 1:MC)
  {
  ZMC <- runif(n,-p,1-p)
   MMC<-0
  for(i in 1:n)
  {
  if(ZMC[i]>0){MMC=MMC+1}
   }
   MMC=MMC/n
   SMC[k]<-sqrt(n)*(MMC-(1-p))/sqrt(p*(1-p))
  }
    pMC<-0
  for (k in 1:MC)
  {
    if(SMC[k]>=S){pMC=pMC+1}
  }
  pMC=pMC/MC
  return(pMC)
}
testZnakowMC(0,50,1, 0.5)
```

```
## [1] 0.052
```

Moc empiryczna w teście Znaków:

```
power_Sign<- function(m, n, sigma,p) {
  num_simulations <- 1000  # liczba symulacji
  power <- 0
  for (i in 1:num_simulations) {
  X <- rnorm(n, m, sigma)
  M <- 0
  for(i in 1:n)
  {
    if(X[i]>0){M=M+1}
  }
  M=M/n
  S<-sqrt(n)*(M-(1-p))/sqrt(p*(1-p))
  P <- 2*(1- pnorm(abs(S),0,1))
  if(P<0.05) power <- power+1
  }
  power <- power/num_simulations
  return(power)
}
power_Sign(0,50,1,0.5)
```

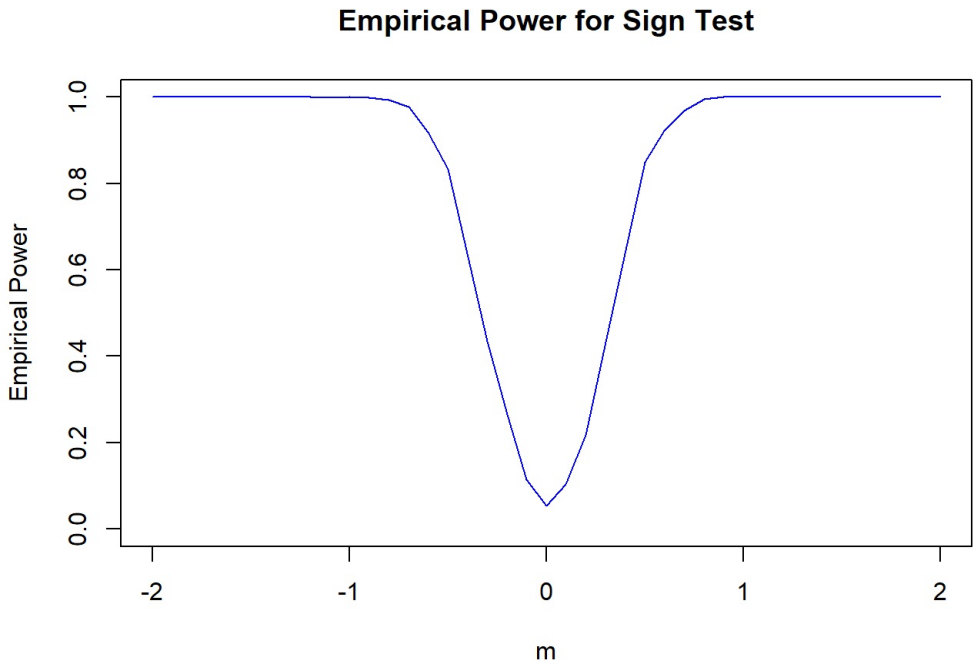
```
## [1] 0.066
```

Wykresy mocy empirycznych dla n = 50, σ = 1,3,6 w zależności od m na przedziale [−2 , 2]

```
m_values <- seq(-2, 2, by = 0.1)
n <- 50
p <- 0.5
```

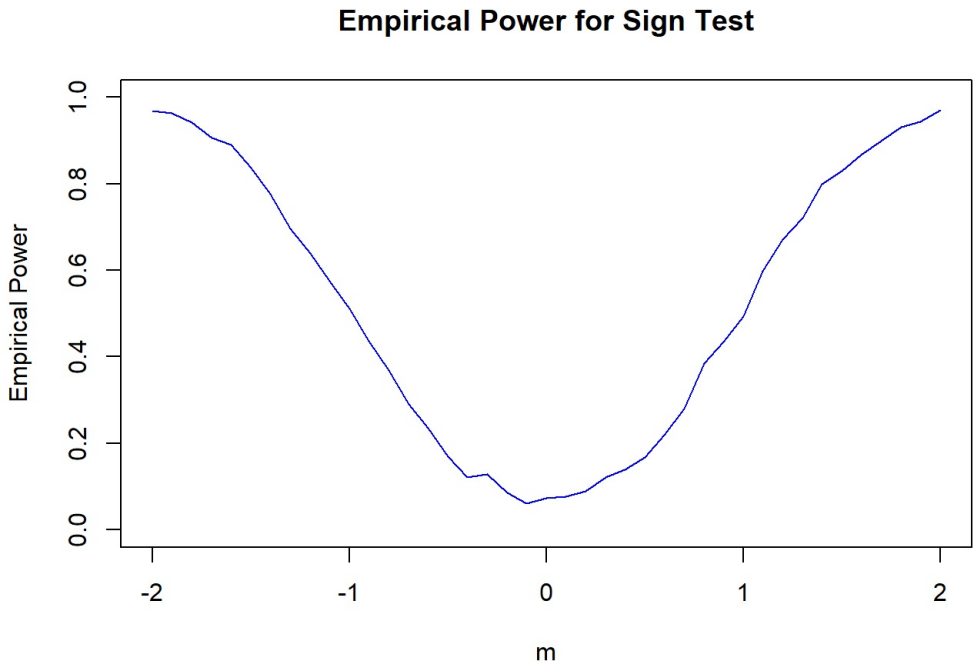
σ = 1

```
sigma <- 1
power_values_S <- sapply(m_values, function(m) power_Sign(m, n, sigma,p))

plot(m_values, power_values_S, type = "l", col = "blue",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power for Sign Test",
     xlim = c(-2, 2), ylim = c(0, 1))
```



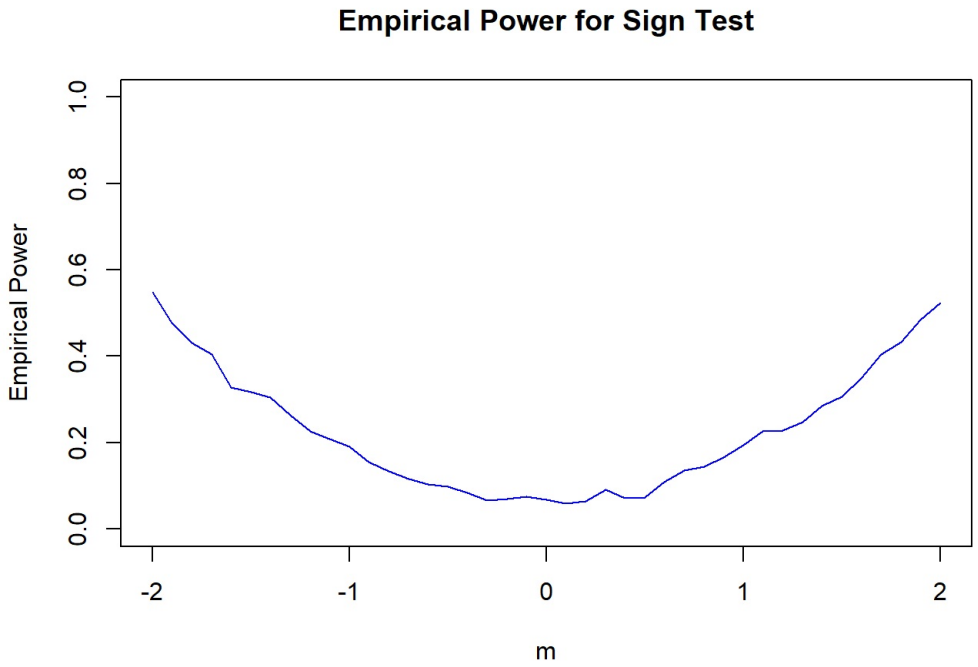
σ = 3

```
sigma <- 3
power_values_S <- sapply(m_values, function(m) power_Sign(m, n, sigma,p))

plot(m_values, power_values_S, type = "l", col = "blue",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power for Sign Test",
     xlim = c(-2, 2), ylim = c(0, 1))
```



σ = 6

```
sigma <- 6
power_values_S <- sapply(m_values, function(m) power_Sign(m, n, sigma,p))

plot(m_values, power_values_S, type = "l", col = "blue",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power for Sign Test",
     xlim = c(-2, 2), ylim = c(0, 1))
```

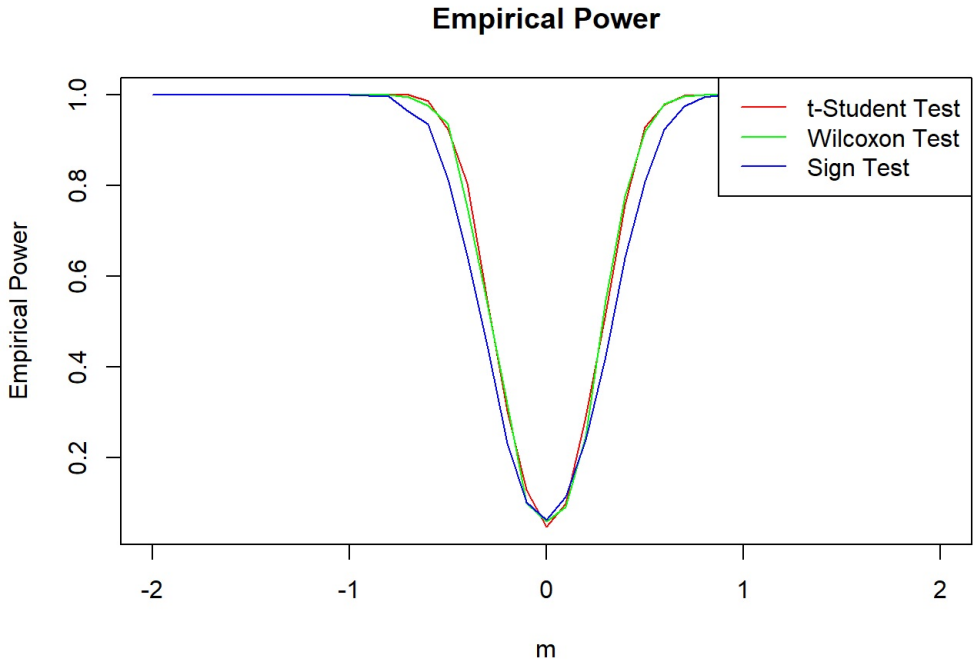


Wykresy końcowe:

```
m_values <- seq(-2, 2, by = 0.1)
n <- 50
p <- 0.5
```

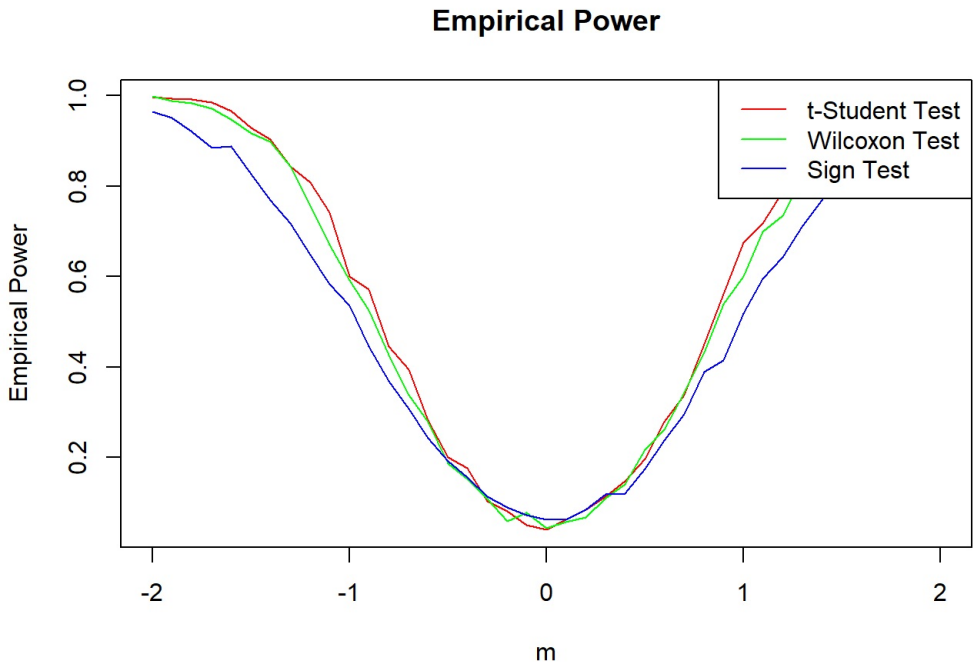
$\sigma = 1$

```
sigma <- 1
power_values1 <- sapply(m_values, function(m) power_tstudent(m, n, sigma))

power_values2 <- sapply(m_values, function(m) power_Wilcoxon(m, n, sigma))

power_values3 <- sapply(m_values, function(m) power_Sign(m, n, sigma,p))

plot(m_values, power_values1, type = "l", col = "red",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power")
lines(m_values, power_values2, type = "l", col = "green")
lines(m_values, power_values3, type = "l", col = "blue")

legend("topright", legend = c("t-Student Test", "Wilcoxon Test", "Sign Test"), col = c("red", "green","blue"), lt
y = 1)
```



$\sigma = 3$

```
sigma <- 3
power_values1 <- sapply(m_values, function(m) power_tstudent(m, n, sigma))

power_values2 <- sapply(m_values, function(m) power_Wilcoxon(m, n, sigma))

power_values3 <- sapply(m_values, function(m) power_Sign(m, n, sigma,p))

plot(m_values, power_values1, type = "l", col = "red",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power")
lines(m_values, power_values2, type = "l", col = "green")
lines(m_values, power_values3, type = "l", col = "blue")

legend("topright", legend = c("t-Student Test", "Wilcoxon Test", "Sign Test"), col = c("red", "green","blue"), lt
y = 1)
```



$\sigma = 6$

```
sigma <- 6
power_values1 <- sapply(m_values, function(m) power_tstudent(m, n, sigma))

power_values2 <- sapply(m_values, function(m) power_Wilcoxon(m, n, sigma))

power_values3 <- sapply(m_values, function(m) power_Sign(m, n, sigma,p))

plot(m_values, power_values1, type = "l", col = "red",
     xlab = "m", ylab = "Empirical Power",
     main = "Empirical Power")
lines(m_values, power_values2, type = "l", col = "green")
lines(m_values, power_values3, type = "l", col = "blue")

legend("topright", legend = c("t-Student Test", "Wilcoxon Test", "Sign Test"), col = c("red", "green","blue"), lt
y = 1)
```