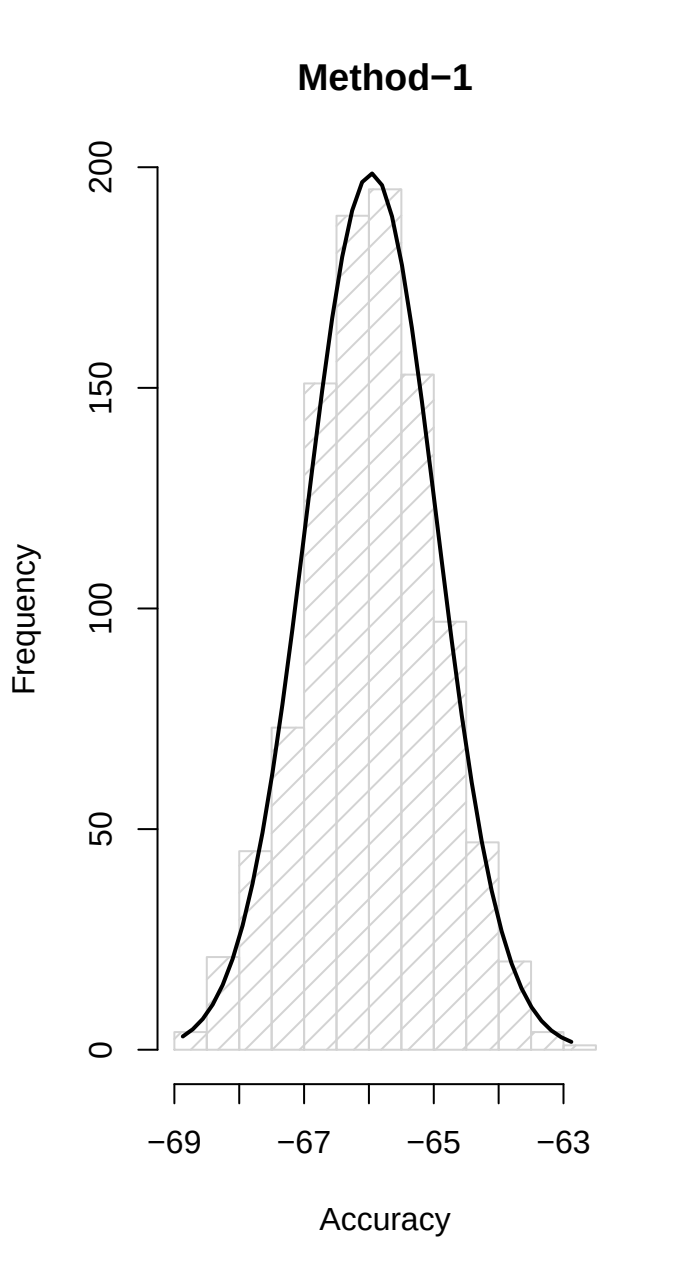
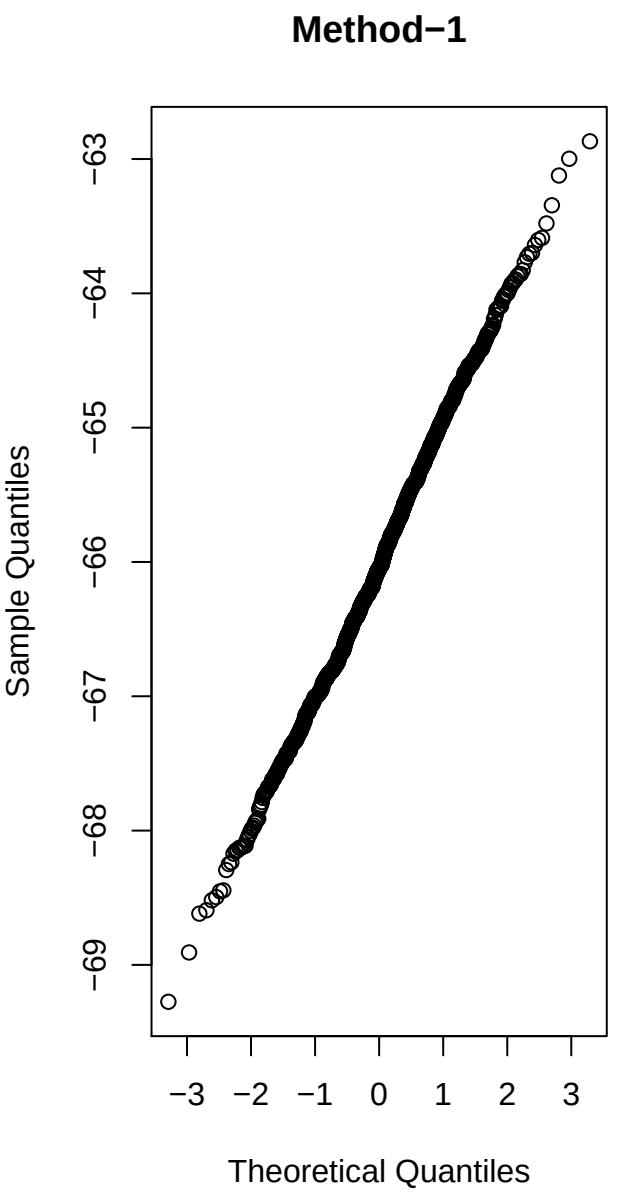
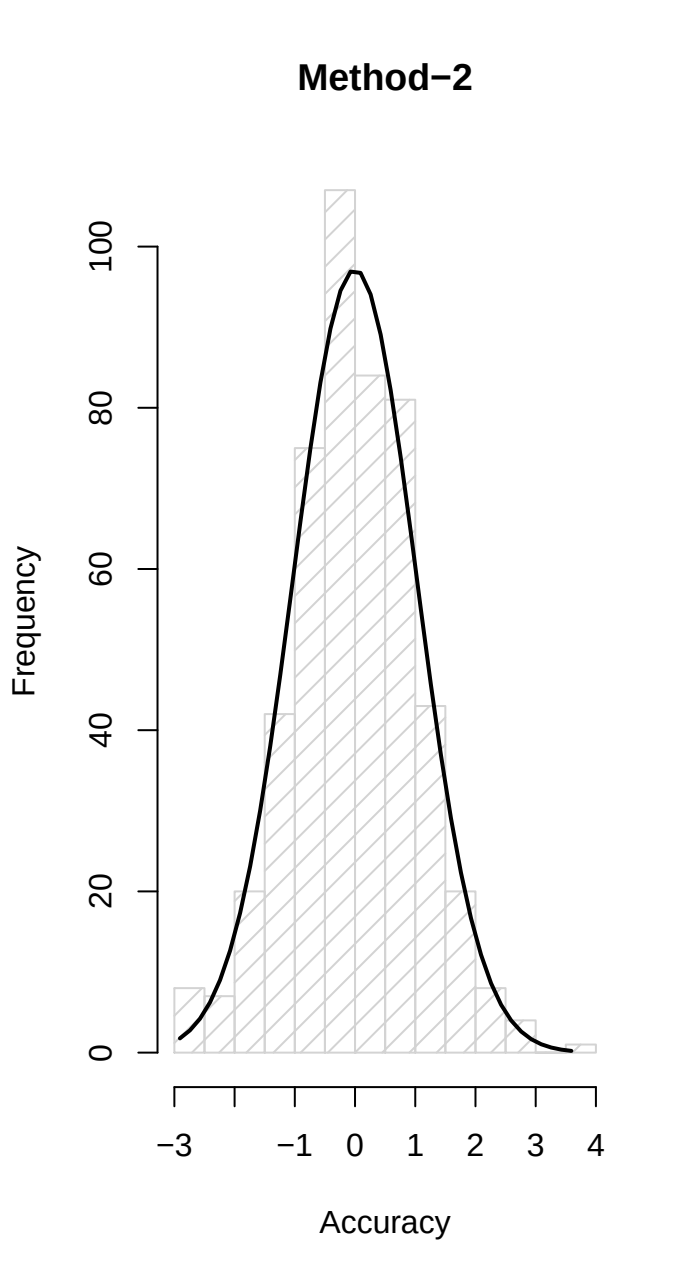
```r
# Author: Jonah Henry
#random.r - generates normal random variables using various methods and uniform random variables


# Method-1: Sum of Uniform Random Variables

Method_1 <- function() {
  data <- runif(12, 0, 1)
  data <- data - 6
  sum(data, na.rm = FALSE)
}

# Method-2: Box-Muller Method

Method_2 <- function() {
  result = c()
  U = runif(2, 0, 1)
  result[1] = sqrt(-2*log(U[1]))*cos(2*pi*U[2])
  result[2] = sqrt(-2*log(U[1]))*sin(2*pi*U[2])
  result
}

# Method-3: Polar Method

Method_3 <- function() {
  S <- 2
  U <- c()
  while(S > 1){
    U <- runif(2, 0, 1)
    U = 2*U-1
    S = U[1]**2+U[2]**2
  }
  c(sqrt(-2*log(S)/S)*U[1], sqrt(-2*log(S)/S)*U[2])

}

#Method-4: Inversion Method

phiInverse <- function(U) {
  w <- sqrt(-2*log(U))
  a <- c(2.515517, 0.802853, 0.010328)
  b <- c(1, 1.432788, 0.189269, 0.001308)
  numerator <- a[1]+a[2]*w+a[3]*(w**2)
  denomenator <- b[1]+b[2]*w+b[3]*(w**2)+b[4]*(w**3)
  -w + numerator/denomenator
}

Method_4 <- function() {
```

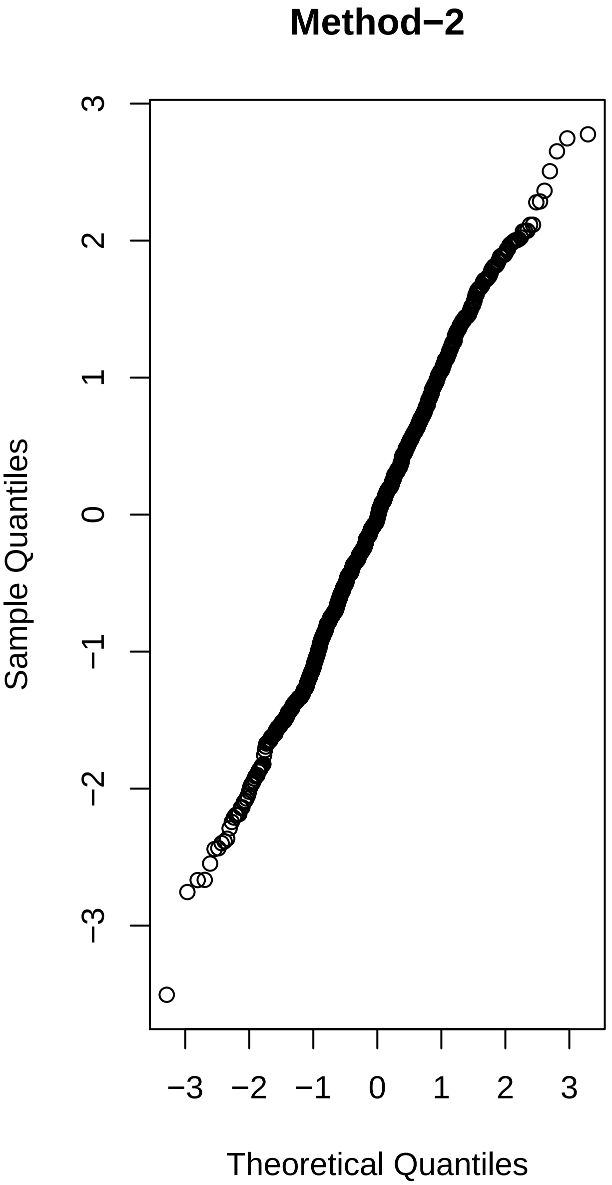
```
  result <- 0
  U <- runif(1,0,1)
  if(U < .5){
    result <- phiInverse(U)
  }
  else{
    result <- phiInverse(1-U)
  }
  result
}

# Method-5: Acceptance-Rejection Method

Method_5 <- function(){
    Z <- 0
    Y <- c(0,0)
    while(Y[2] < .5*(Y[1]-1)**2) {
        U <- runif(2,0,1)
        Y <- -log(U)
    }
    Z <- Y[1]
    U <- runif(1,0,1)
    if(U <= .5){
        Z <- abs(Z)
    }
    else{
        Z <- -abs(Z)
    }
    Z
}

# Method-6: Using Generalized Exponential Distribution

Method_6 <- function() {
    U <- runif(1, 0, 1)
    X <- -log(1 - U ** .0775)
    (log(X) - 1.0821) / .3807
}

# Method-7: Bol'shev Formula

Method_7 <- function() {
    data <- runif(5, 0, 1)
    data <- sqrt(3) * (2 * data - 1)
    X <- sum(data, na.rm = FALSE)
    X <- X / sqrt(5)
    X - .01 * (3 * X - X ** 3)
}
```

```r
# Method-8: Inversion Method

Method_8 <- function() {
    U <- runif(1, 0, 1)
    (1/1.702)*(-log(1/U-1))
}

# Method-9: Proposed Method

Method_9 <- function() {
    U <- runif(1, 0, 1)
    X1 <- tanh(-31.35694 + 28.77154 * U)
    X2 <- tanh(-2.57136 - 31.16364 * U)
    X3 <- tanh(3.94963 - 1.66888 * U)
    X4 <- tanh(2.31229 + 1.84289 * U)
    .46615 + 90.72192 * X1 - 89.36967 * X2 - 96.55499 * X3 + 97.36346 * X4
}


Main <- function() {

    #Plotting done here
    #Methods 1, 3-9 run 1000 times to generate 1000 variables
    #Method 2 runs 500 times to generate 1000 variables (returns 2 variables each
      time)

}

Main()
```

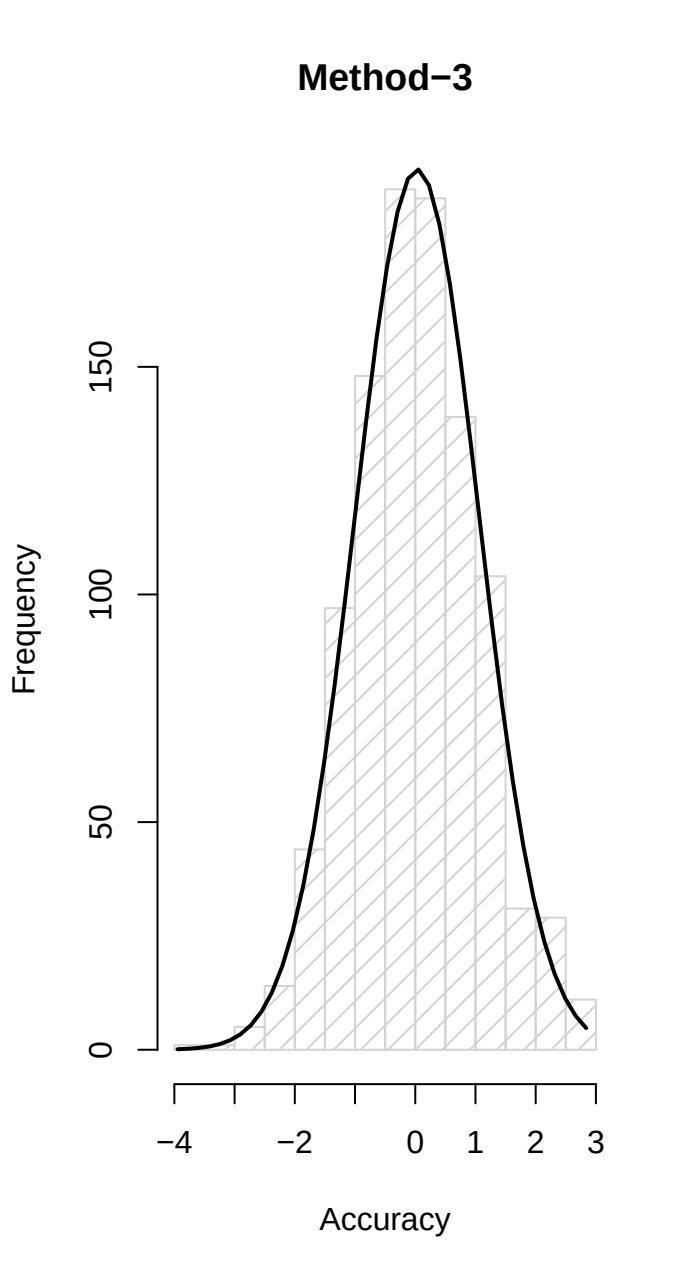

Method−1
Frequency
0
50
100
150
200
−69
−67
−65
−63
Accuracy

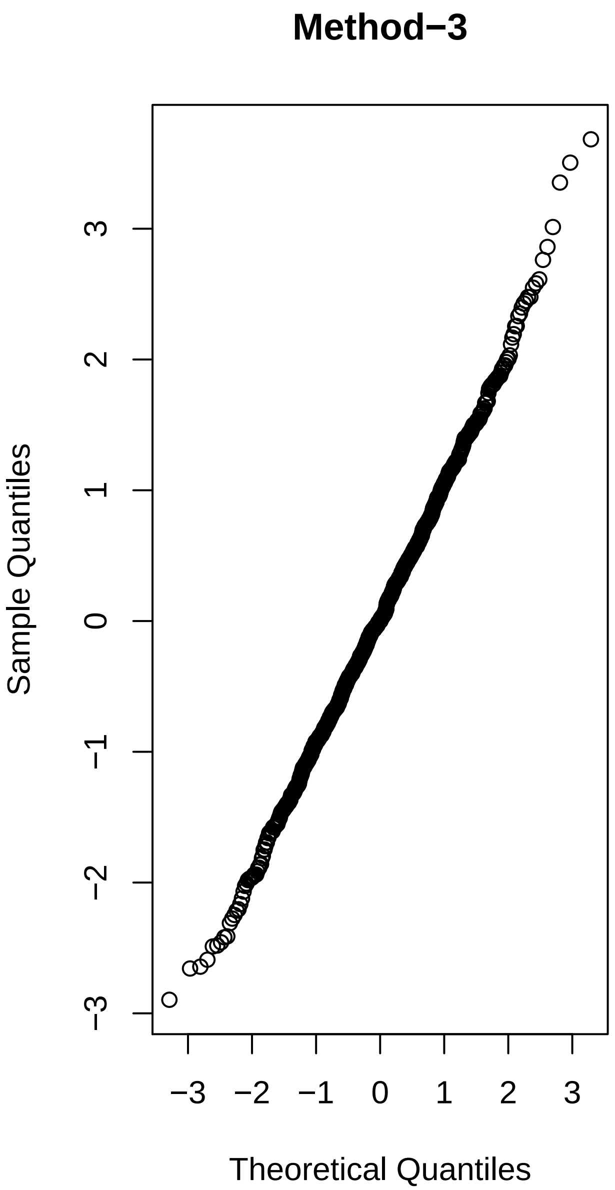

Method−1
Sample Quantiles
−63
−64
−65
−66
−67
−68
−69
−3
−2
−1
0
1
2
3
Theoretical Quantiles

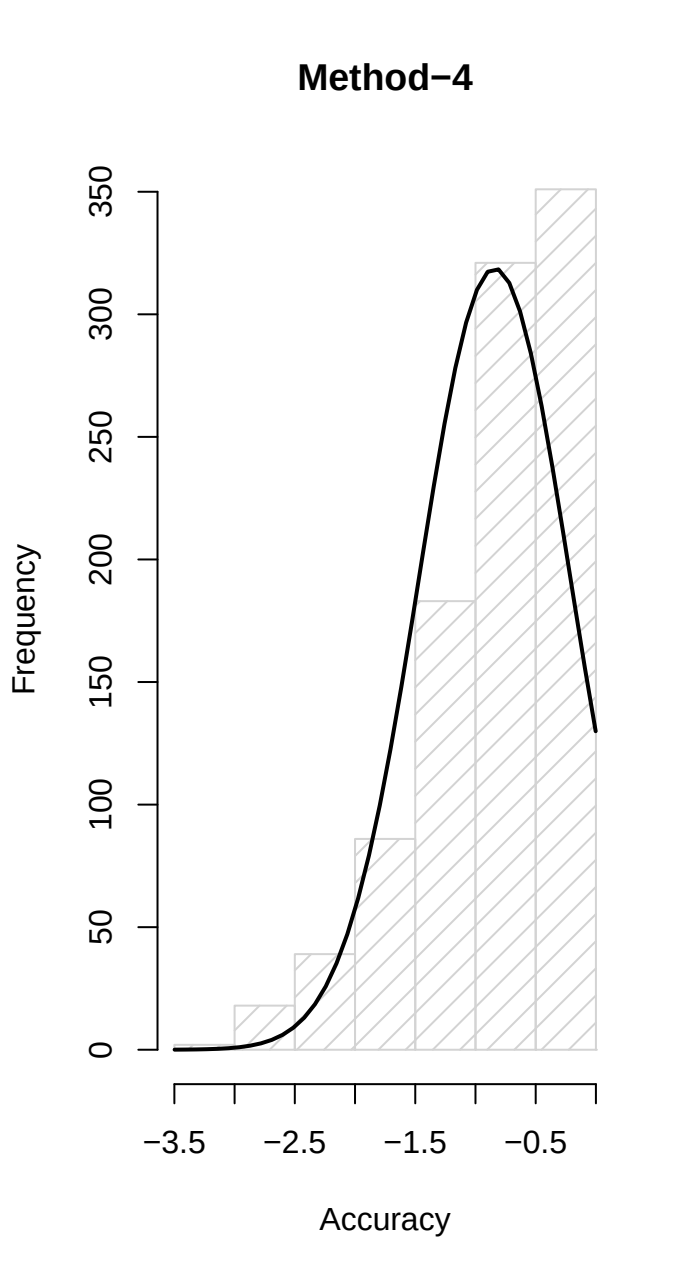

Method−2
Frequency
0
20
40
60
80
100
−3
−1
0
1
2
3
4
Accuracy

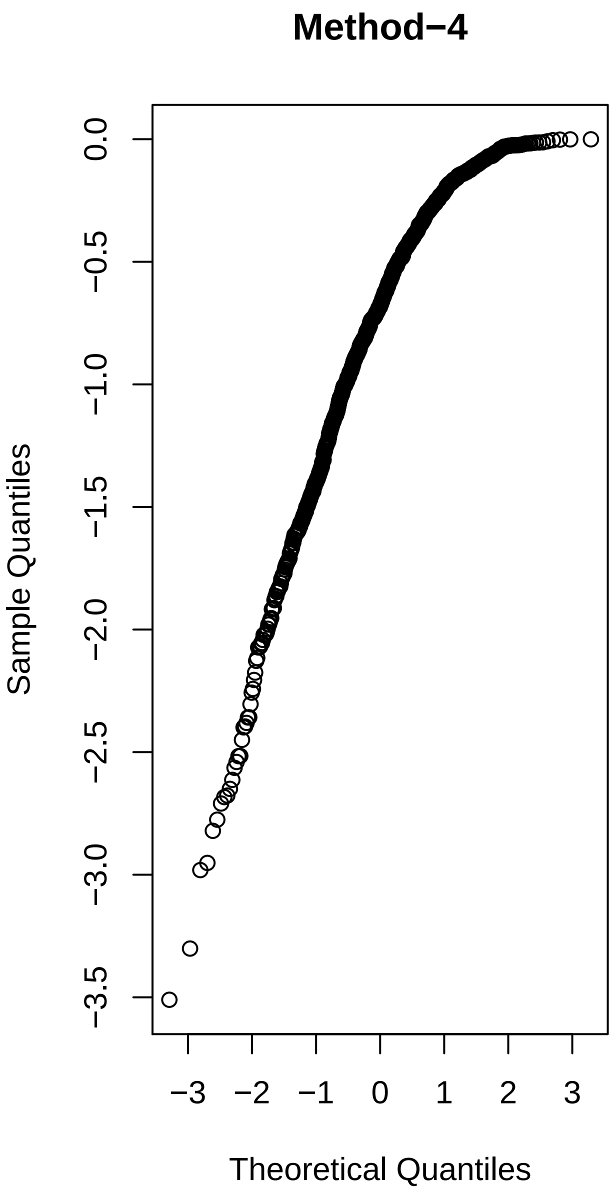


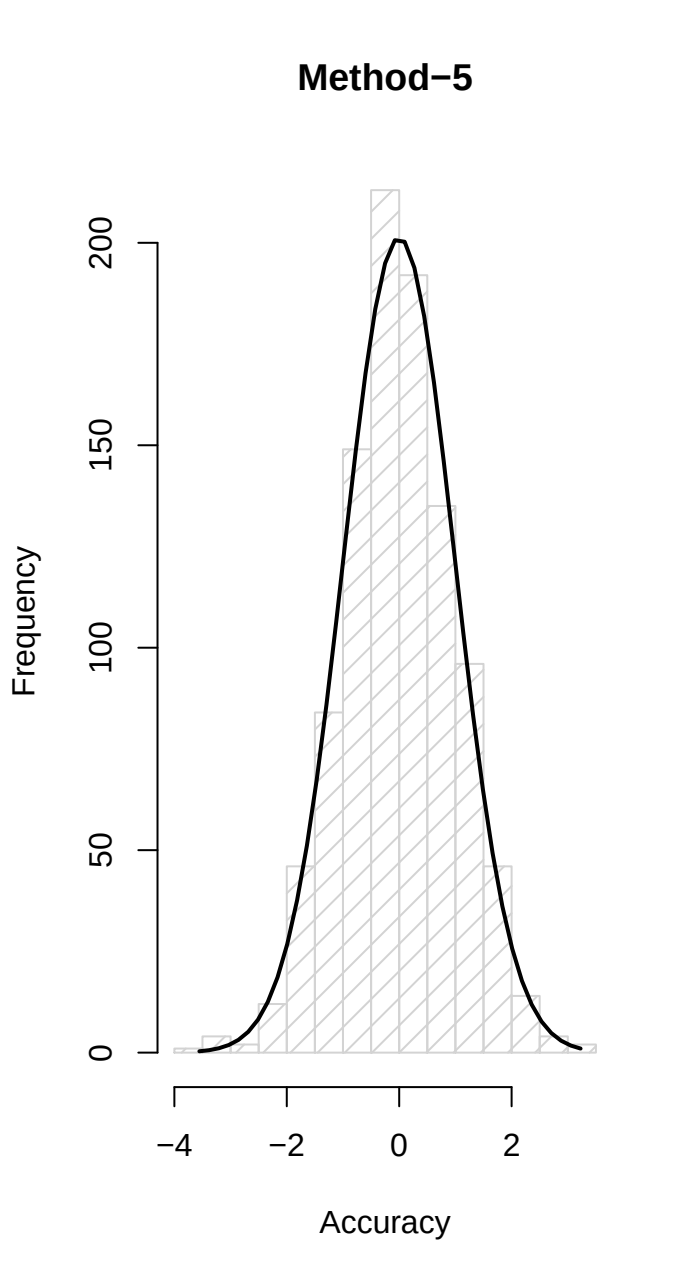
Method−2
Sample Quantiles
Theoretical Quantiles
3
2
1
0
−1
−2
−3
−3 −2 −1 0 1 2 3

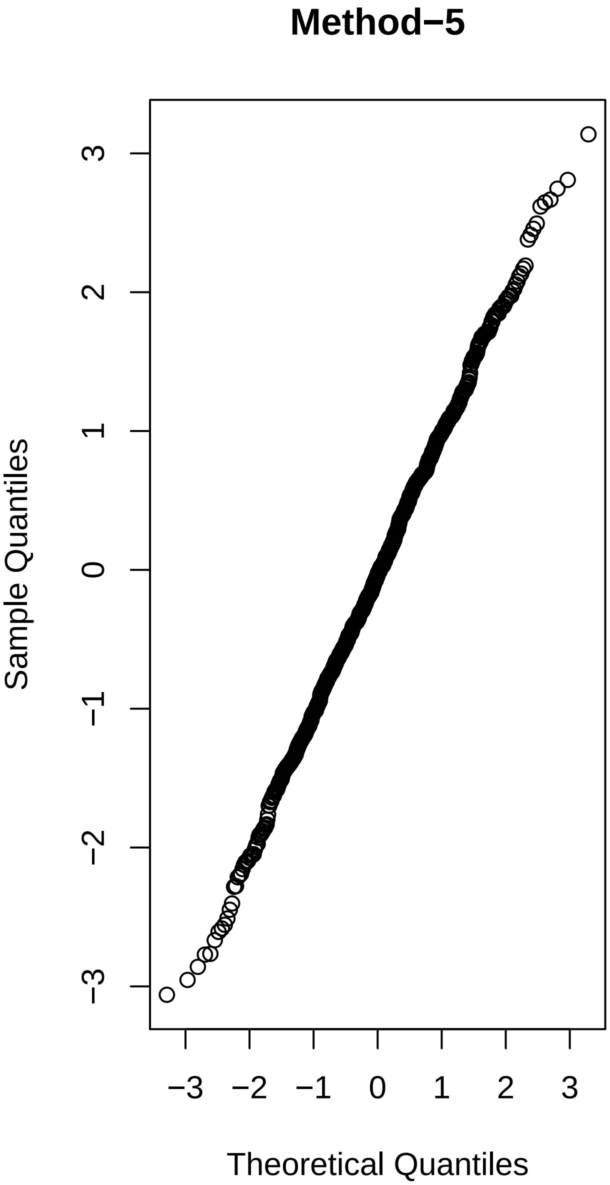

Method−3
Frequency
0
50
100
150
−4
−2
0
1
2
3
Accuracy


Method−3
Sample Quantiles
Theoretical Quantiles
−3
−2
−1
0
1
2
3

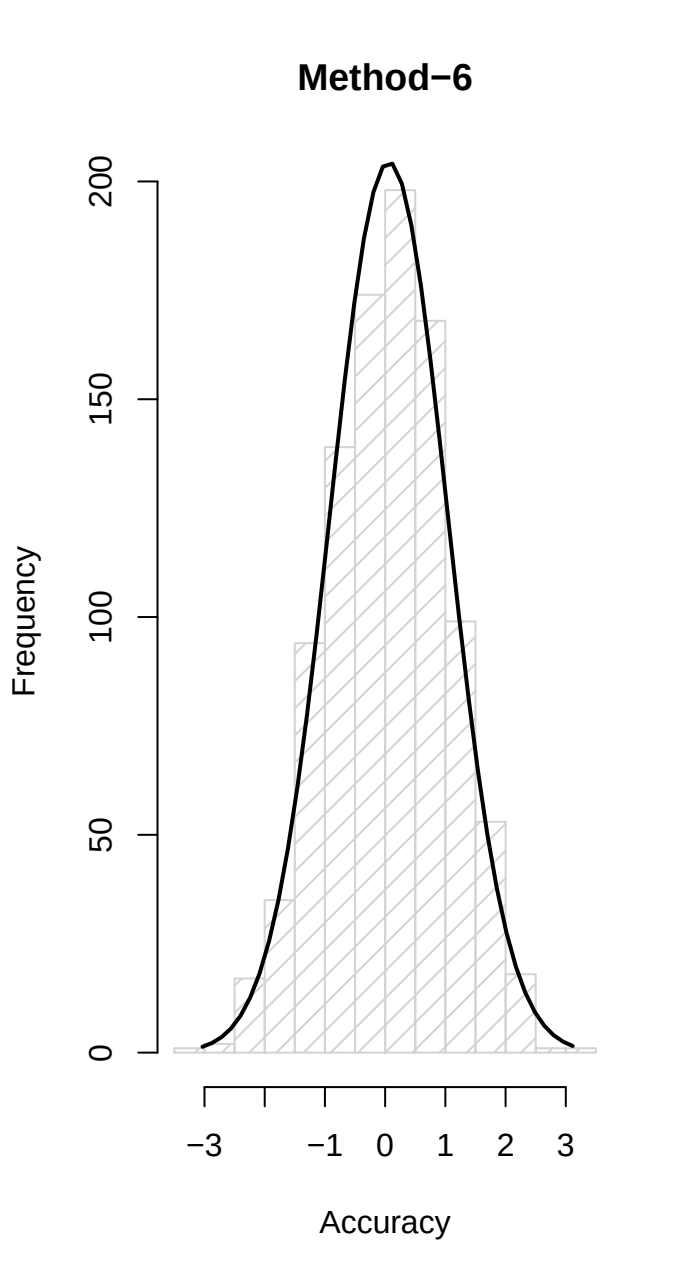

Method−4
Frequency
0
50
100
150
200
250
300
350
−3.5
−2.5
−1.5
−0.5
Accuracy

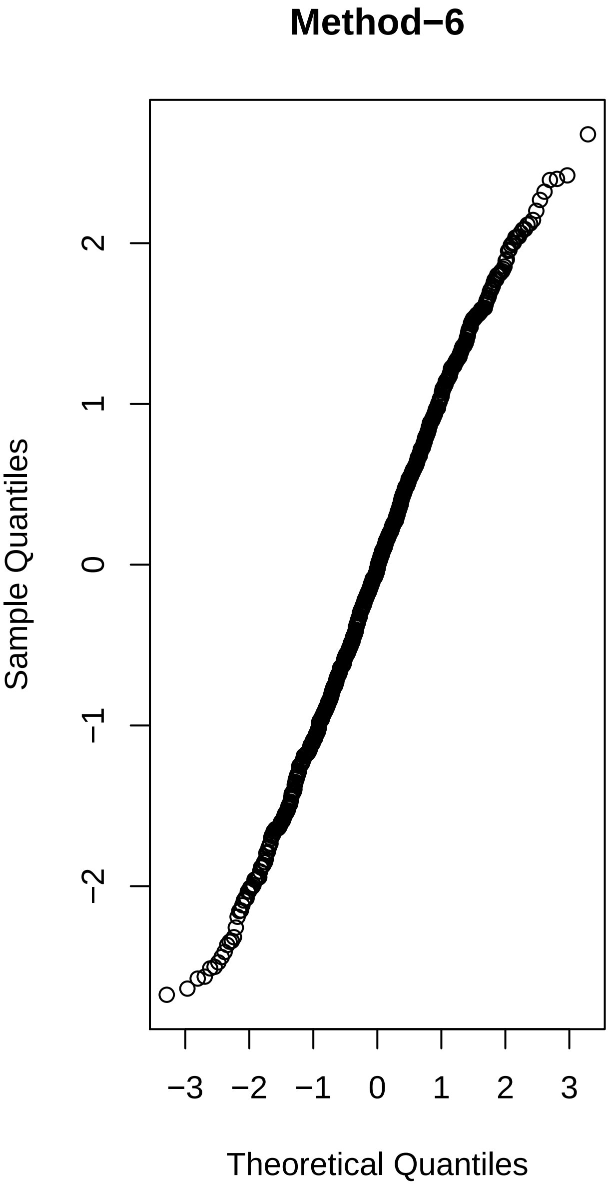

Method−4
Sample Quantiles
Theoretical Quantiles
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
−3
−2
−1
0
1
2
3

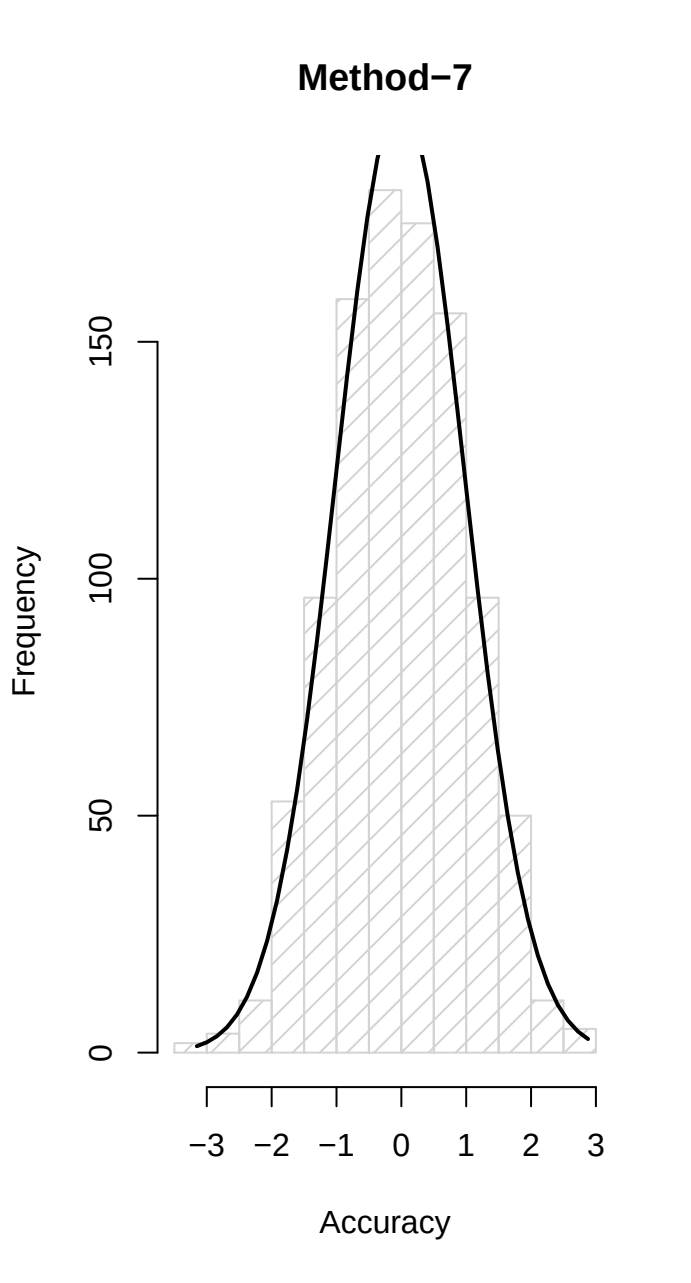

Method−5
Frequency
0
50
100
150
200
−4
−2
0
2
Accuracy

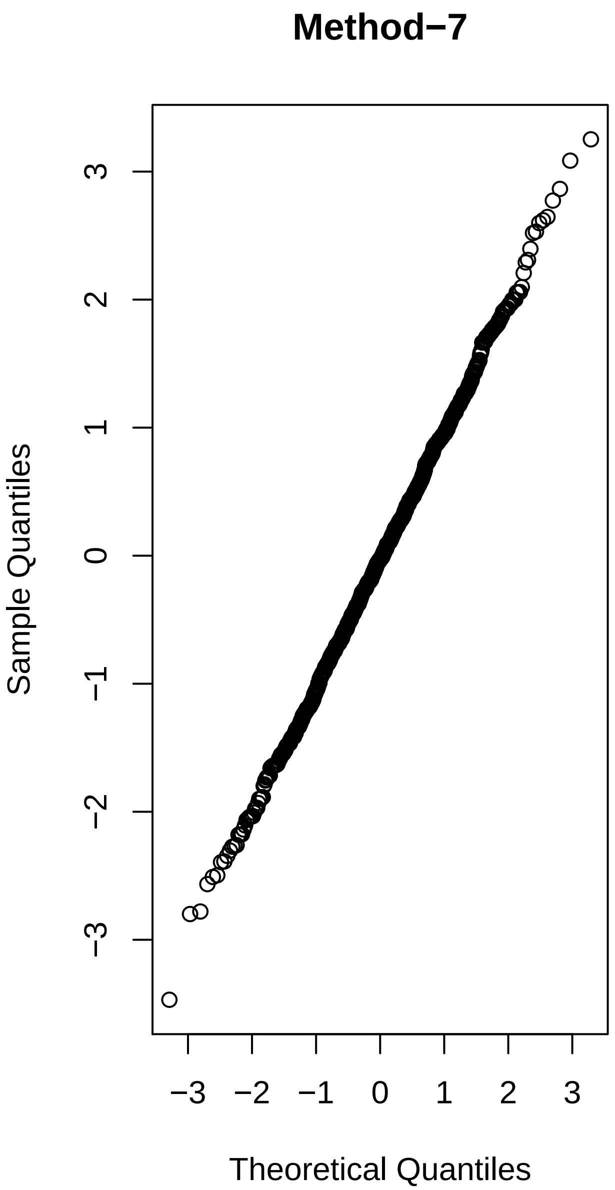


Method−5
Sample Quantiles
Theoretical Quantiles
3
2
1
0
−1
−2
−3
−3 −2 −1 0 1 2 3

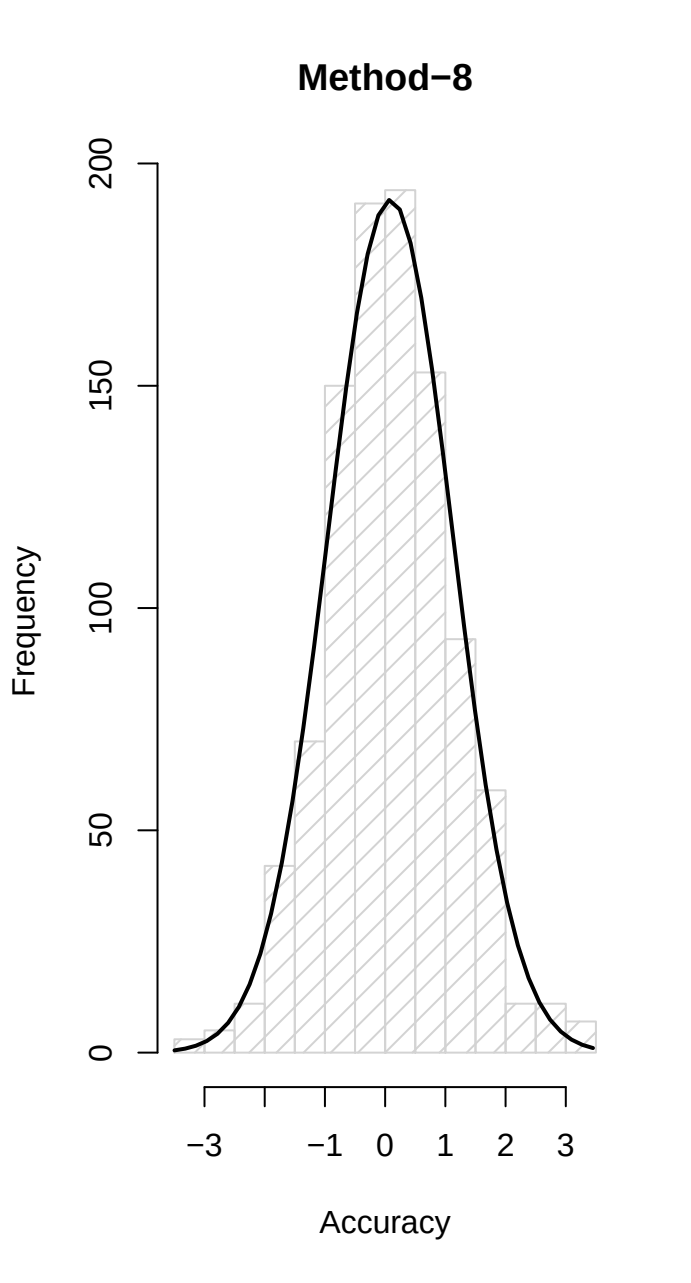

Method−6
Frequency
0
50
100
150
200
−3
−1
0
1
2
3
Accuracy


Method−6
Sample Quantiles
Theoretical Quantiles
−2
−1
0
1
2
−3
−2
−1
0
1
2
3

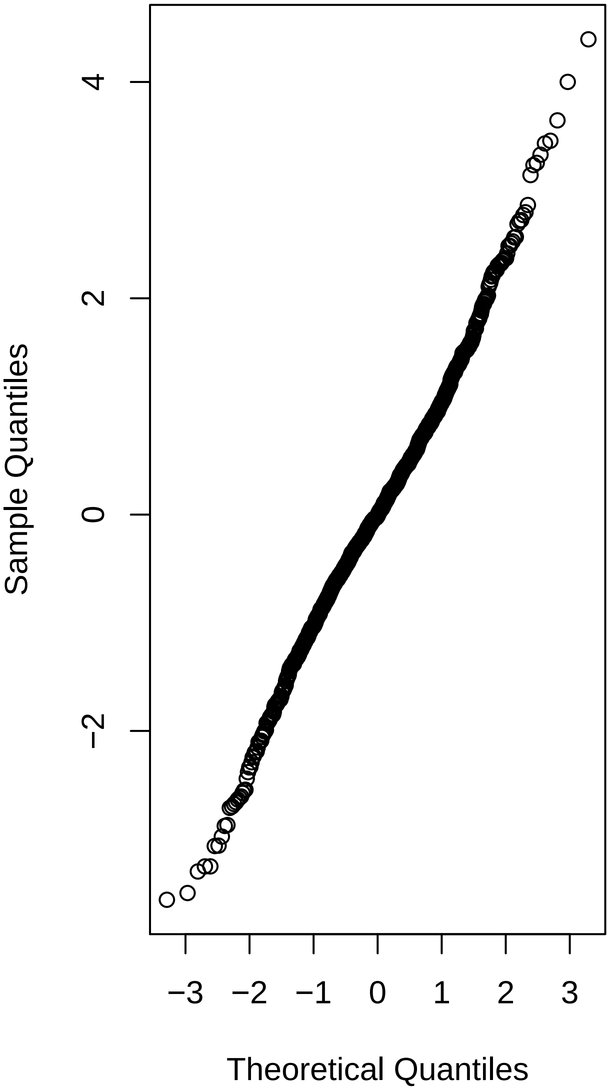

Method−7
Frequency
0
50
100
150
−3
−2
−1
0
1
2
3
Accuracy

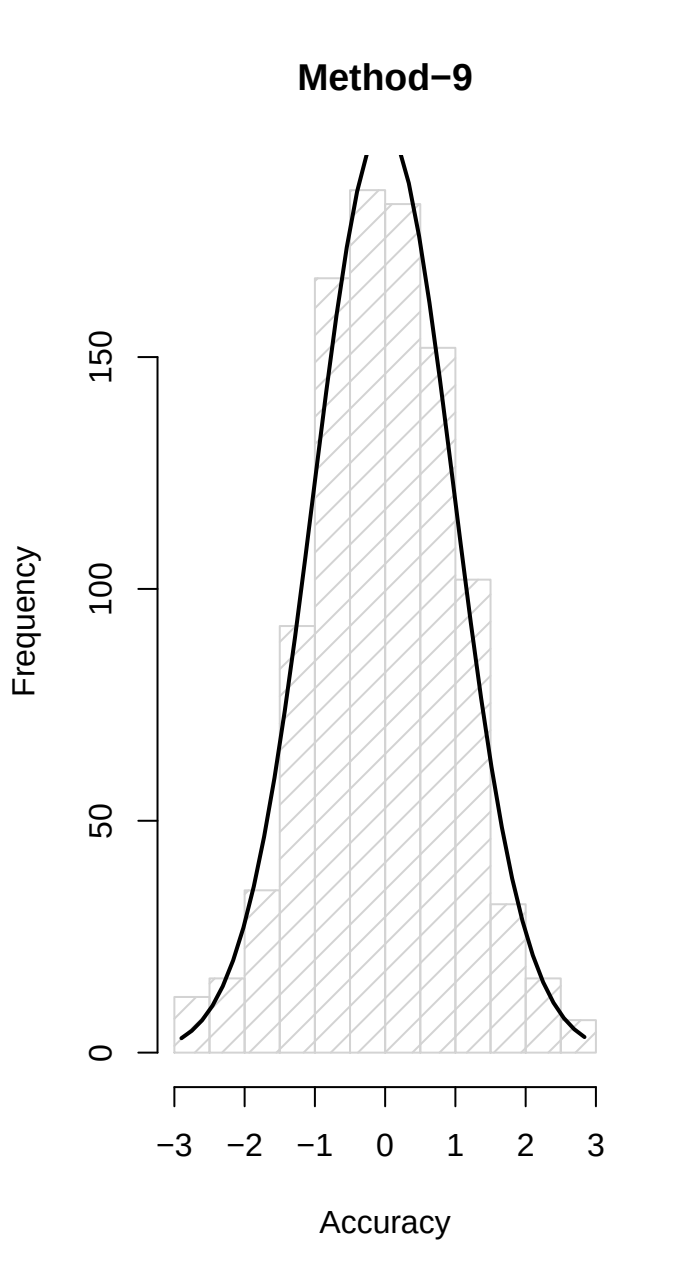


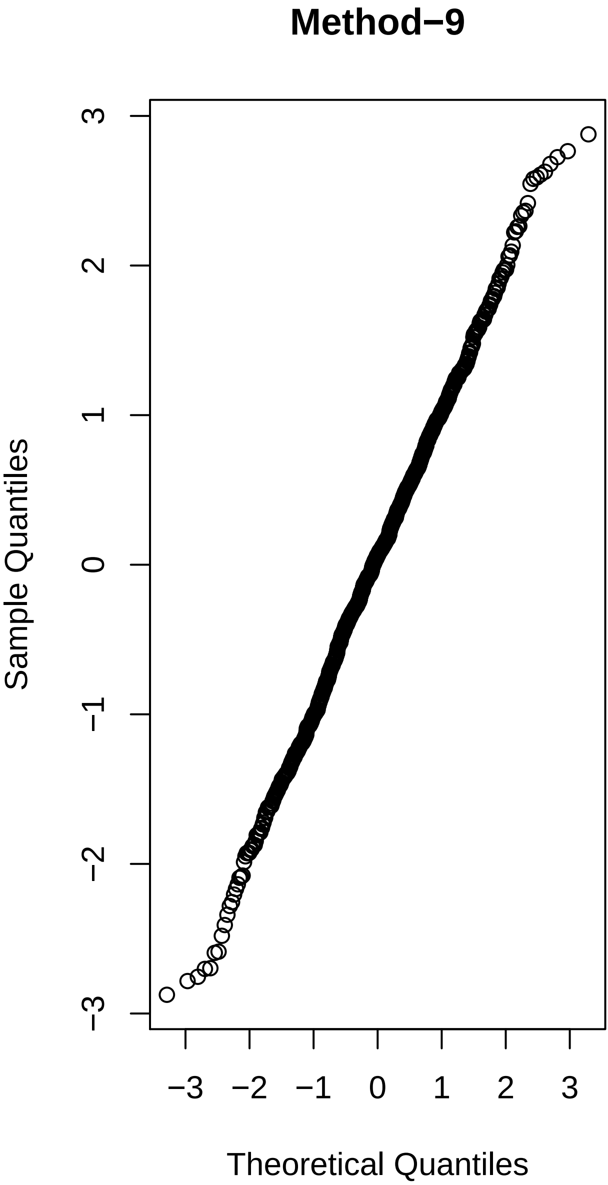
Method−7
Sample Quantiles
Theoretical Quantiles
−3
−2
−1
0
1
2
3

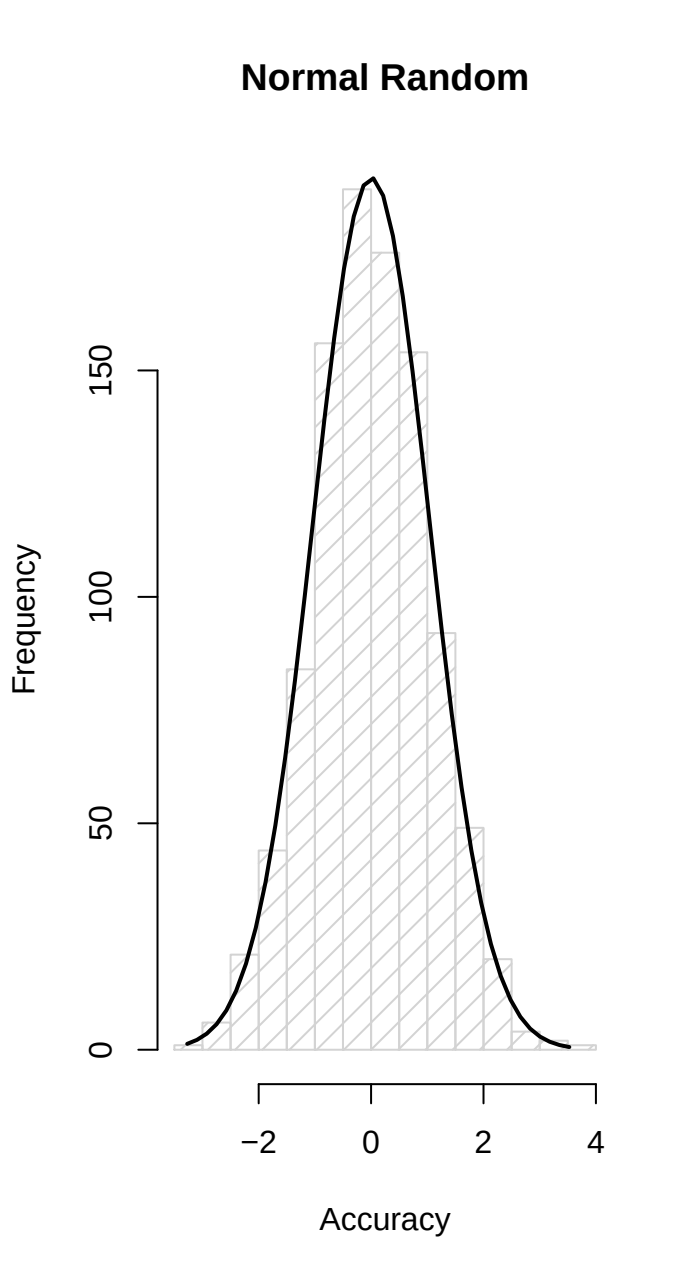

Method−8
Frequency
0
50
100
150
200
−3
−1
0
1
2
3
Accuracy

Method−8




Method−9
Frequency
0
50
100
150
−3
−2
−1
0
1
2
3
Accuracy


Method−9
Sample Quantiles
Theoretical Quantiles
−3
−2
−1
0
1
2
3


Normal Random
Frequency
0
50
100
150
−2
0
2
4
Accuracy