HW-6_AK

Akhil

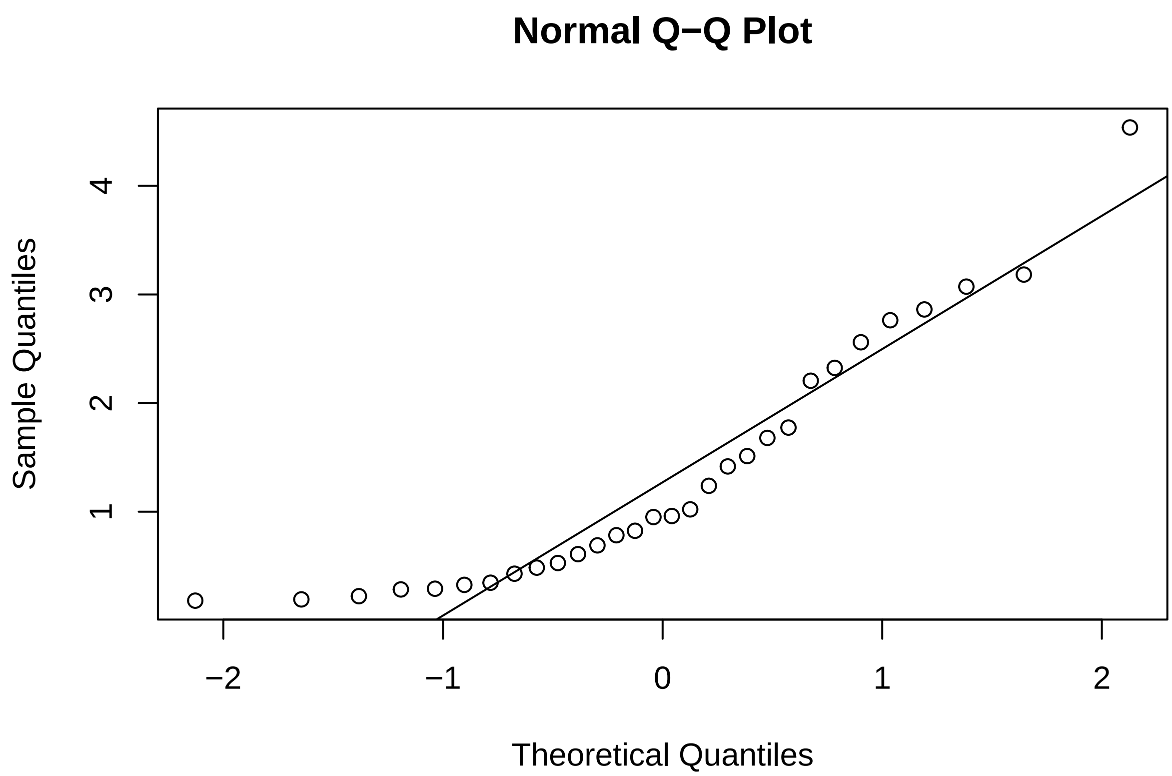
2024-10-11

R Markdown

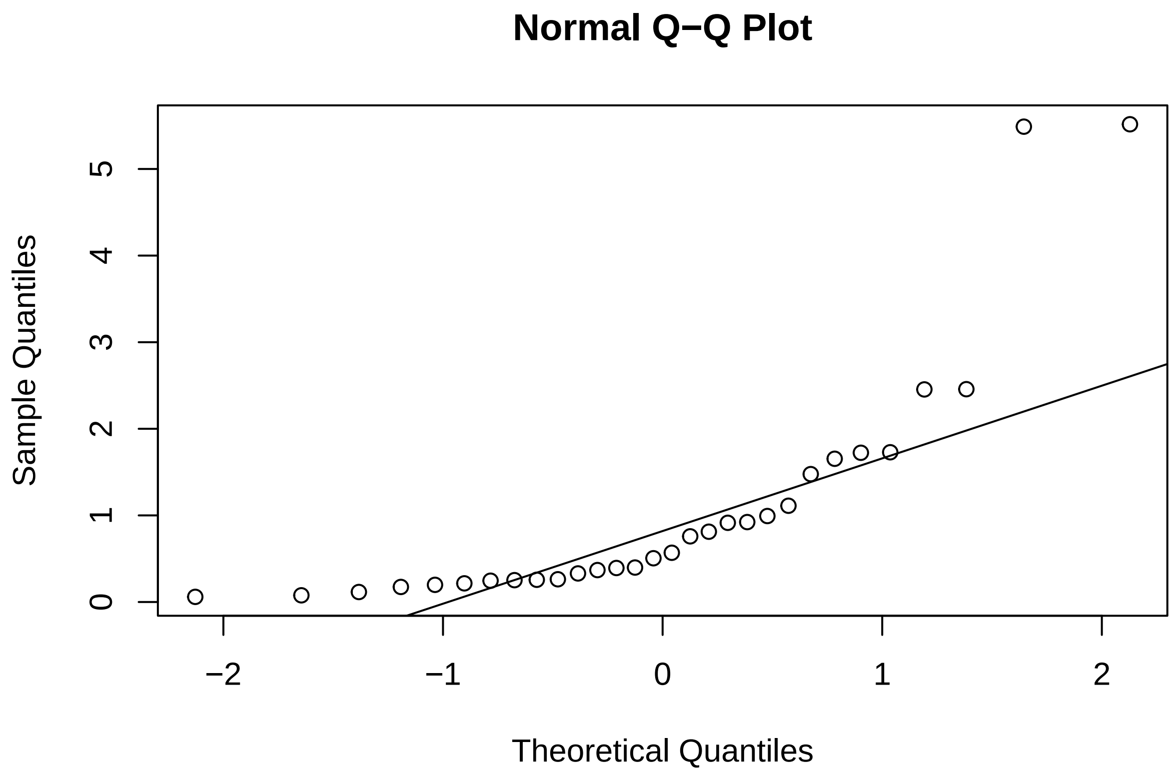
```
BoxMuller <- function(n, mu = 0, sigma = 1) {
  u1 <- runif(n / 2)
  u2 <- runif(n / 2)

  z1 <- sqrt(-2 * log(u1)) * cos(2 * pi * u2)
  z2 <- sqrt(-2 * log(u1)) * sin(2 * pi * u2)

  # Combine z1 and z2 and scale by mean (mu) and standard deviation (sigma)
  z <- c(z1, z2) # Combine two vectors to create n normal variates
  normal_variates <- mu + sigma * z
  return(normal_variates)
}

PolarMethod <- function(n, mu = 0, sigma = 1) {
  normals <- numeric(0) # Initialize empty vector to store normal variates
  while (length(normals) < n) {
    u1 <- runif(1, -1, 1)
    u2 <- runif(1, -1, 1)
    s <- u1^2 + u2^2

    if (s > 0 && s < 1) {
      z1 <- u1 * sqrt(-2 * log(s) / s)
      z2 <- u2 * sqrt(-2 * log(s) / s)
      normals <- c(normals, mu + sigma * z1, mu + sigma * z2) # Add to result
    }
  }
  return(normals[1:n]) # Return exactly n normal variates
}

n <- 30
mu <- 0
sigma <- 1

boxmuller_samples <- BoxMuller(n, mu, sigma)
polar_samples <- PolarMethod(n, mu, sigma)

ks_test_result <- ks.test(boxmuller_samples, polar_samples)
print(ks_test_result)
```

```
##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  boxmuller_samples and polar_samples
## D = 0.23333, p-value = 0.3929
## alternative hypothesis: two-sided
```

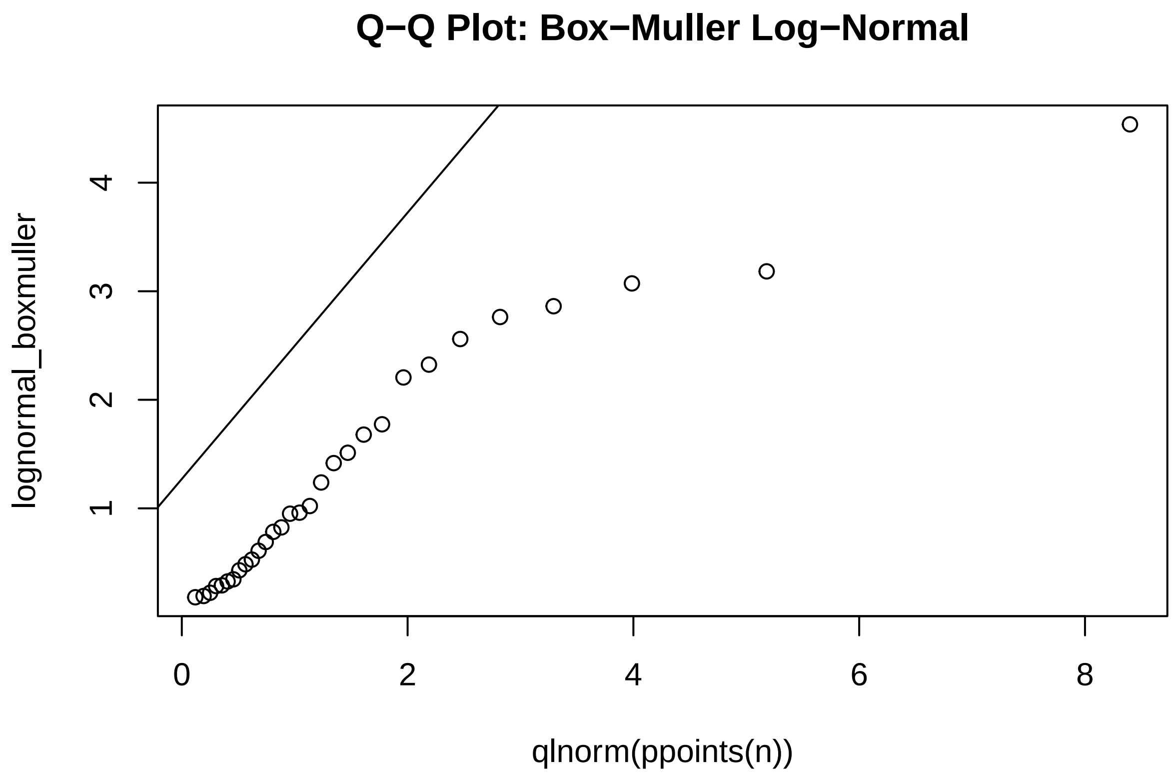
```r
lognormal_boxmuller <- exp(boxmuller_samples)
lognormal_polar <- exp(polar_samples)

# Q-Q plot for log-normal distribution (using Box-Muller generated log-normal variates)
qqnorm(lognormal_boxmuller)
qqline(lognormal_boxmuller)
```



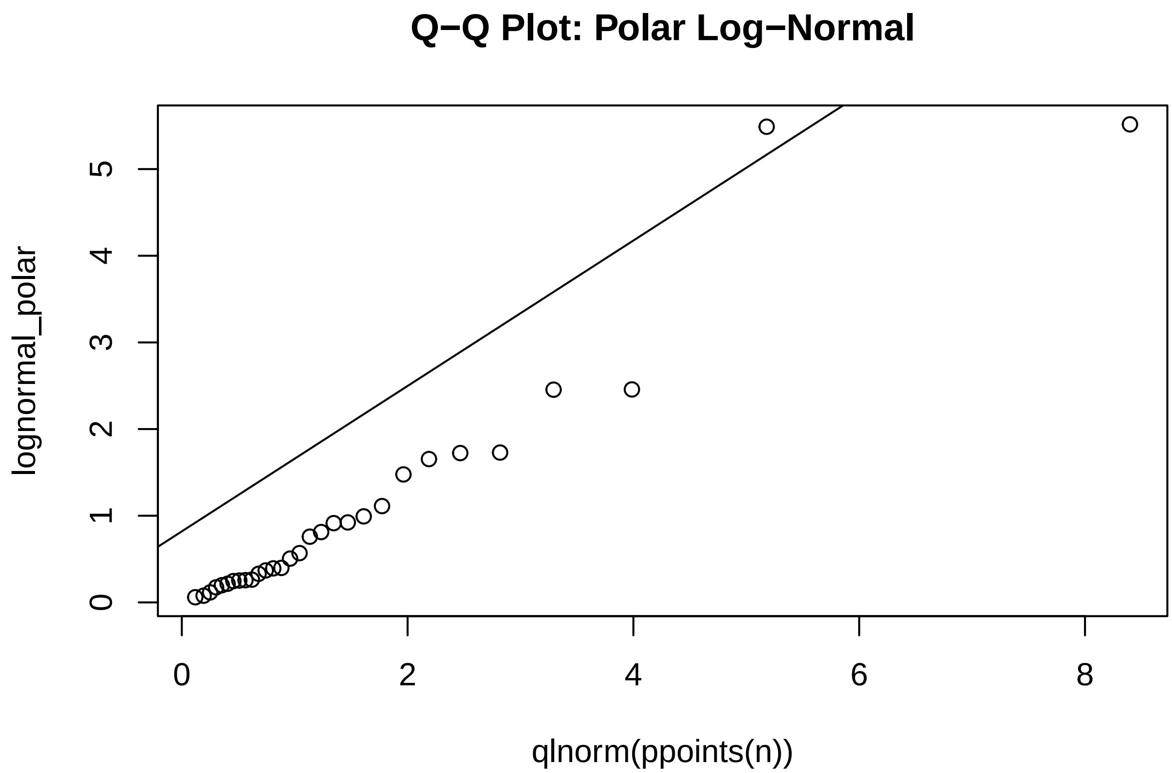

```r
# Q-Q plot for log-normal distribution (using Polar method generated log-normal variates)
qqnorm(lognormal_polar)
qqline(lognormal_polar)
```




```r
# Q-Q plot against theoretical log-normal quantiles
qqplot(qlnorm(ppoints(n)), lognormal_boxmuller, main="Q-Q Plot: Box-Muller Log-Normal")
qqline(lognormal_boxmuller)
```




```
qqplot(qlnorm(ppoints(n)), lognormal_polar, main="Q-Q Plot: Polar Log-Normal")
qqline(lognormal_polar)
```

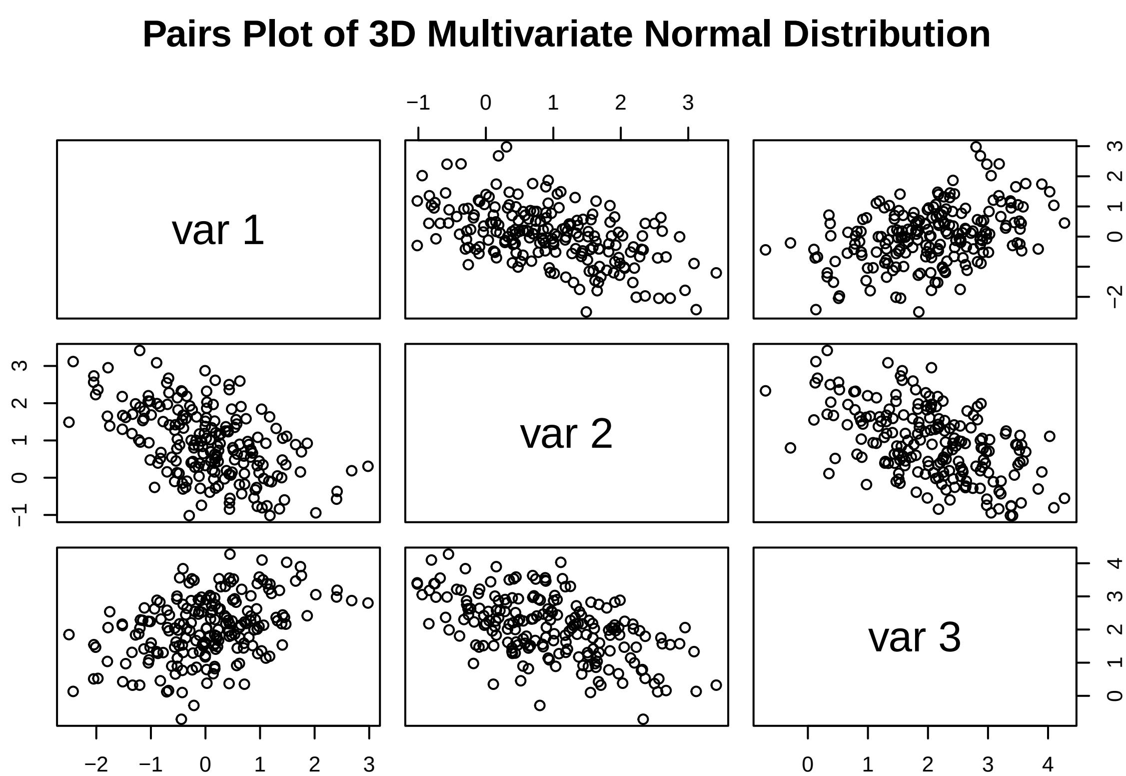



```
mymvrnorm <- function(n, mu, Sigma) {
  p <- length(mu)

  eig <- eigen(Sigma)
  A <- eig$vectors %*% diag(sqrt(eig$values)) %*% t(eig$vectors)

  Z <- matrix(0, nrow = n, ncol = p)

  for (i in 1:n) {
    for (j in seq(1, p, by = 2)) {
      u1 <- runif(1)
      u2 <- runif(1)
      z1 <- sqrt(-2 * log(u1)) * cos(2 * pi * u2)
      z2 <- sqrt(-2 * log(u1)) * sin(2 * pi * u2)
      Z[i, j] <- z1
      if (j + 1 <= p) {
        Z[i, j + 1] <- z2
      }
    }
  }

  Y <- matrix(0, nrow = n, ncol = p)
  for (i in 1:n) {
    Y[i, ] <- mu + A %*% Z[i, ]
  }
```

```r
  return(Y)
}

mu <- c(0, 1, 2)
Sigma <- matrix(c(1.0, -0.5, 0.5,
                  -0.5, 1.0, -0.5,
                  0.5, -0.5, 1.0), nrow = 3, byrow = TRUE)

set.seed(123)
samples <- mymvrnorm(200, mu, Sigma)

head(samples)
```

```
##            [,1]       [,2]     [,3]
## [1,]  0.9500032 -0.7680897 3.385341
## [2,]  0.5013511  0.8062659 2.887023
## [3,] -1.1269701  1.5916635 1.414957
## [4,] -0.2487615  0.4137752 3.530221
## [5,]  1.7585731  0.6936964 3.628668
## [6,]  0.1686668  0.3907384 2.956012
```

```r
pairs(samples, main = "Pairs Plot of 3D Multivariate Normal Distribution")
```



```r
library(stats)

g1 <- function(U, a) {
  Phi_a <- pnorm(a)
  qnorm(Phi_a + (1 - Phi_a) * U)
}

g2 <- function(U, a) {
  Phi_neg_a <- pnorm(-a)
  -qnorm(Phi_neg_a * (1 - U))
}

U_vals <- (seq(1, 1000) - 0.5) / 1000

find_failure <- function(g_function, a_start = 1) {
  a <- a_start
  while (TRUE) {
    results <- g_function(U_vals, a)
    if (any(is.nan(results) | is.infinite(results))) {
      return(a)
    }
    a <- a + 1
  }
}

a_g1_fail <- find_failure(g1, 2)
cat("The smallest a for which g1 fails is:", a_g1_fail, "\n")
```

```
## The smallest a for which g1 fails is: 8
```

```r
a_g2_fail <- find_failure(g2, 2)
cat("The smallest a for which g2 fails is:", a_g2_fail, "\n")
```

```
## The smallest a for which g2 fails is: 38
```

```r
r <- 5
p <- 0.4

theoretical_mean <- r * (1 - p) / p
theoretical_variance <- r * (1 - p) / p^2

n <- 1000

theta <- (1 - p) / p
lambda <- rgamma(n, shape = r, scale = theta)

nb_samples <- rpois(n, lambda)

sample_mean <- mean(nb_samples)
sample_variance <- var(nb_samples)

cat("Sample Mean: ", sample_mean, "\n")
```

```
## Sample Mean:  7.36
```

```r
cat("Sample Variance: ", sample_variance, "\n")
```

```
## Sample Variance:  16.84925
```

```r
cat("Theoretical Mean: ", theoretical_mean, "\n")
```

```
## Theoretical Mean:  7.5
```

```r
cat("Theoretical Variance: ", theoretical_variance, "\n")
```

```
## Theoretical Variance:  18.75
```

```r
cat("Difference in Mean: ", abs(sample_mean - theoretical_mean), "\n")
```

```
## Difference in Mean:  0.14
```

```r
cat("Difference in Variance: ", abs(sample_variance - theoretical_variance), "\n")
```

```
## Difference in Variance:  1.900751
```