Assignment-2

Abheek Ghosh
140123047

August 14, 2017

Question 1

Code for R

```
rm(list=ls())
d = read.table("d-csp0108.txt", header=TRUE)

names = c('C', 'SP')

for (k in 2:3) {
   mean_d = mean(d[,k])
   sd_d = sd(d[,k])
   kurtosis_d = mean( (d[,k] - mean_d)^4 / sd_d^4 )
   print(sprintf("Kurtosis of %s = %f", names[k-1], kurtosis_d))
}
```

Kurtosis of C = 75.040894

Kurtosis of SP = 13.235003

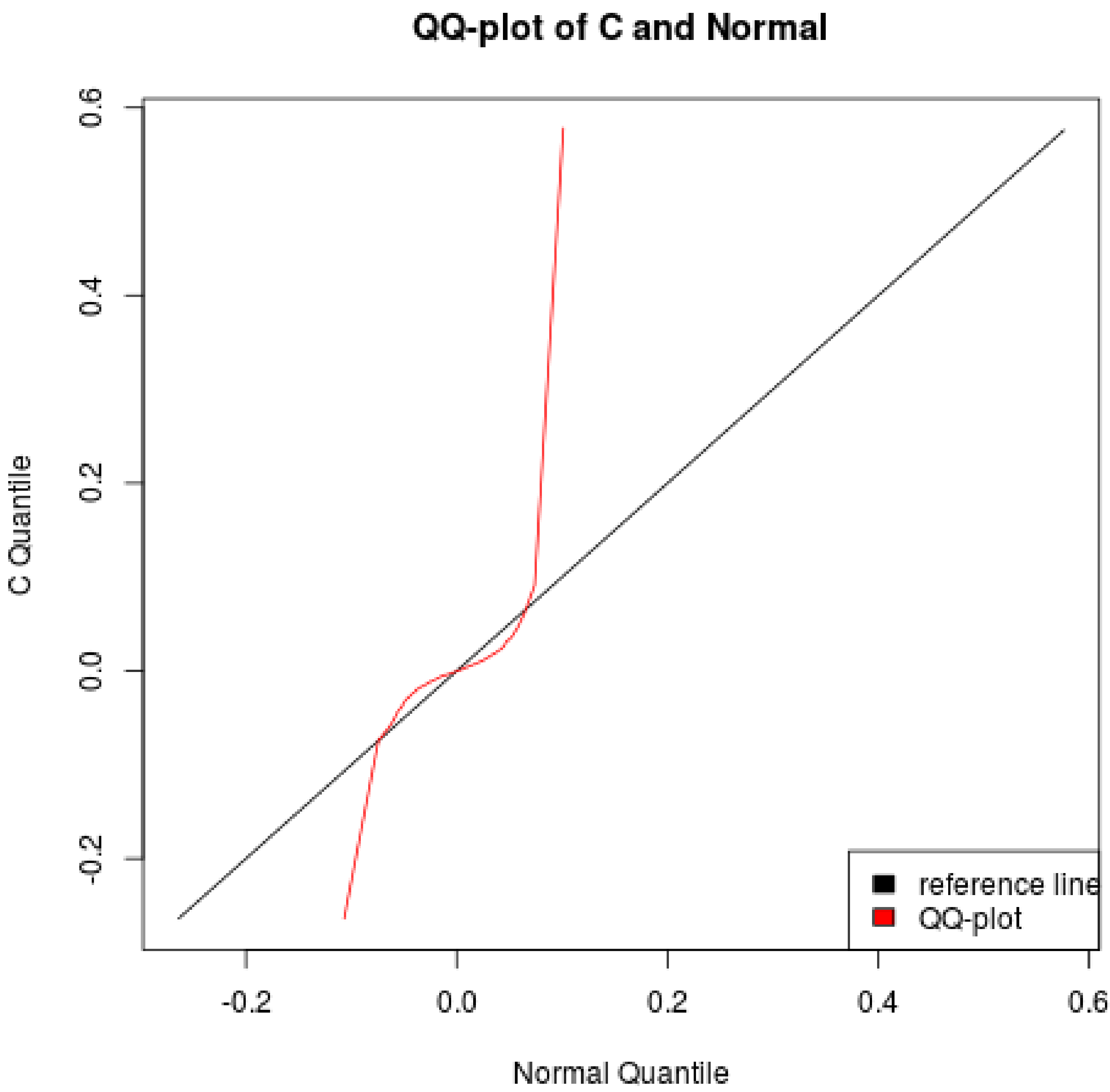
From the Kurtosis values we observe that C is more heavy tailed than SP. Both of them are very heavy tailed than normal.

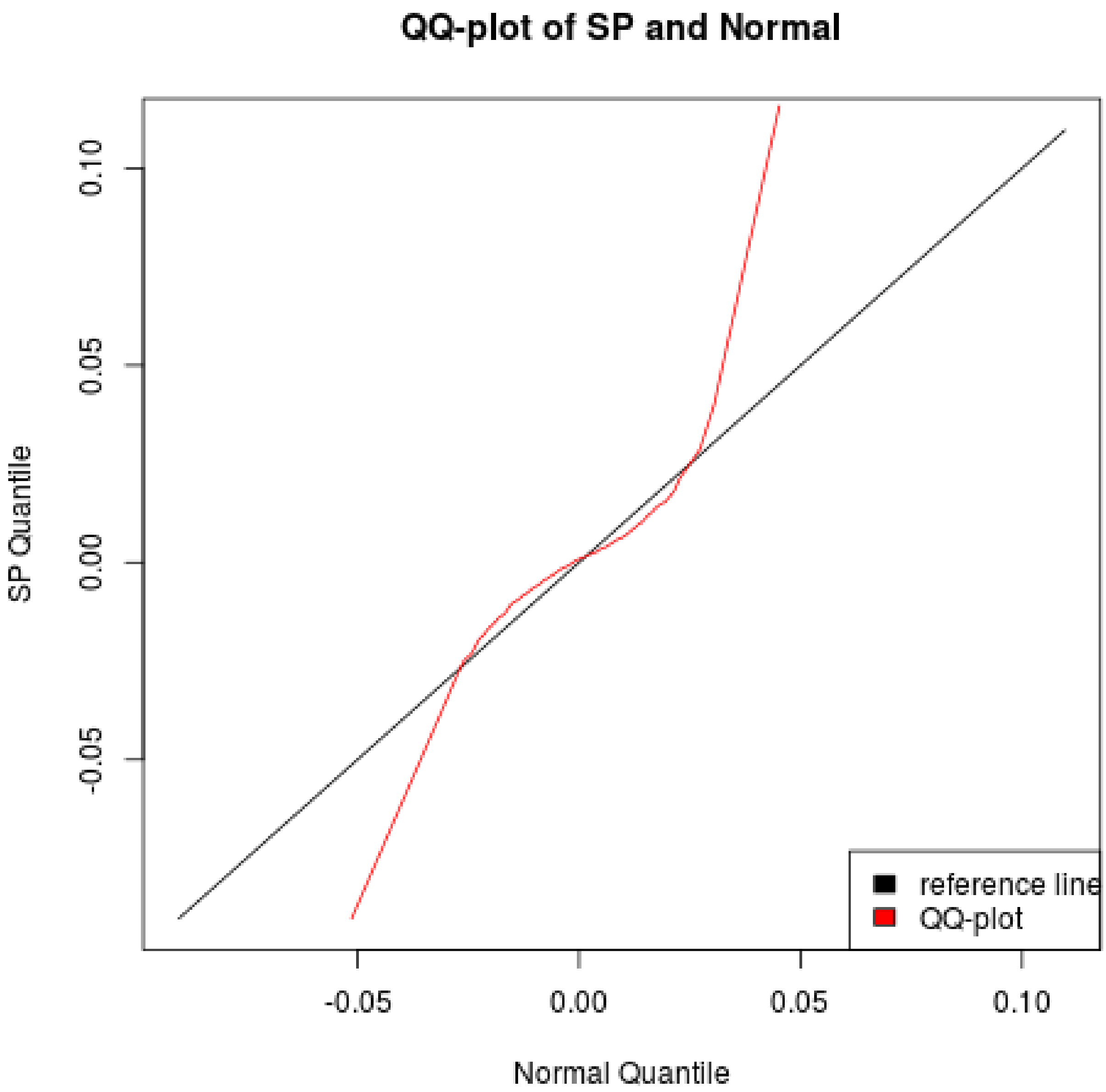
Question 2

Code for R

```
rm(list=ls())
d = read.table("d-csp0108.txt", header=TRUE)

names = c('C', 'SP')

for (k in 2:3) {
   p = seq(0, 1, 0.01)
```

```r
    Q_data = quantile(d[,k], probs = p)
    Q_norm = quantile(rnorm(length(d[,k]), mean = mean(d[,k]), sd = sd(d[,k])), probs = p)
    # Q_norm = qnorm(p, mean = mean(d[,k]), sd = sd(d[,k]))

    refLine = seq(min(Q_data), max(Q_data), 0.01)
    plot(refLine, refLine, main = sprintf("QQ-plot of %s and Normal", names[k-1]),
        xlab = "Normal Quantile", ylab = sprintf("%s Quantile", names[k-1]), col = 'black', type
            = 'l')
    lines(Q_norm, Q_data, col = 'red')
    legend("bottomright", legend = c('reference line', 'QQ-plot'), fill = c('black', 'red'))

    dev.copy(png, sprintf("plots/plotb%d.png", k-1))
    dev.off()
}
```


QQ-plot of C and Normal
C Quantile
Normal Quantile
-0.2
0.0
0.2
0.4
0.6
reference line
QQ-plot


QQ-plot of SP and Normal
SP Quantile
0.10
0.05
0.00
-0.05
-0.05
0.00
0.05
0.10
Normal Quantile
reference line
QQ-plot

From the QQ-plots we observe that both C and SP are much heavy tailed than normal distribution.

Question 3

Code for R

```
rm(list=ls())

shape = 2
scale = 1
count = c(20, 40, 100, 200)


for (k in count) {
  d = rweibull(k, shape = shape, scale = scale)
  print(sprintf("For %d samples, maximum = %f", k, max(d)))
}
```

We obsereve that in the individual sample sets, there is a unique maximum value. Among the 4 sets of different sizes the maximum element changes.

For 20 samples, maximum = 2.035289

For 40 samples, maximum = 1.674097

For 100 samples, maximum = 2.400425

For 200 samples, maximum = 2.223440

Question 4

Code for R

```
rm(list=ls())

shape = 2
scale = 1
count = c(20, 40, 100, 200)

log_likelihood <- function(beta, theta, X)
{
  n = length(X)
  temp1 = sum(log(X))
  temp3 = X^beta
  temp3 = vector(,n)
  for(i in 1:n)
  temp3[i] = X[i]^beta
  temp3 = sum(temp3)
  return(n*log(beta) + n*beta*log(theta) + (beta-1)*temp1 - (theta^beta)*temp3 )
```

```
}

log_likelihood_beta <- function(beta, X)
{
    n = length(X)
    temp1 = log(X)
    temp3 = X^beta
    temp2 = sum(temp1*temp3)
    temp1 = sum(temp1)
    temp3 = sum(temp3)
    return(n/beta - n*temp2/temp3 + temp1)
}

log_likelihood_beta_prime <- function(beta, X)
{
    n = length(X)
    temp1 = log(X)
    temp3 = X^beta
    temp2 = temp1*temp3
    temp4 = (temp1*temp1) %*% temp3 # scalar
    temp2 = sum(temp2)
    temp3 = sum(temp3)
    return( -n/(beta^2) -n*( (temp2^2 - temp3*temp4)/(temp3^2) ) )
}

newton_raphson <- function(f, f_prime, X, tol=1e-5, x0=1, N=100)
{
    i=1
    x1 = x0
    p = numeric(N)
    while (i<=N)
    {
        df.dx = (f(x0 + tol,X) - f(x0,X))/tol
        x1 = x0 - (f(x0,X)/df.dx)
        p[i] = x1
        i = i + 1
        if(abs(x1 - x0) < tol)
            break
        x0 = x1
    }
    return(x0)
}

estimateParam <- function(X)
{
# beta0 = uniroot(function(x) log_likelihood_beta(x,X),lower = 1, upper = 5, tol = 1e-5)$root
    beta0 = newton_raphson(log_likelihood_beta,log_likelihood_beta_prime, X)
    temp3 = sum(X^beta0)
```

```
    theta0 = (length(X)/temp3)^(1/beta0)
    return(c(beta0, theta0))
}

for (k in count) {
    param = estimateParam(rweibull(k, shape, scale))
    cat(sprintf("For %d samples, estimates of shape = %f, scale = %f\n", k, param[1], 1/param
        [2]))
}
```

For 20 samples, estimates of shape = 2.018674, scale = 1.024464, mean square error = 0.021763

For 40 samples, estimates of shape = 1.988225, scale = 1.023521, mean square error = 0.018600

For 100 samples, estimates of shape = 1.874818, scale = 1.003445, mean square error = 0.088550

For 200 samples, estimates of shape = 1.993557, scale = 0.986334, mean square error = 0.010683