Assignment-1

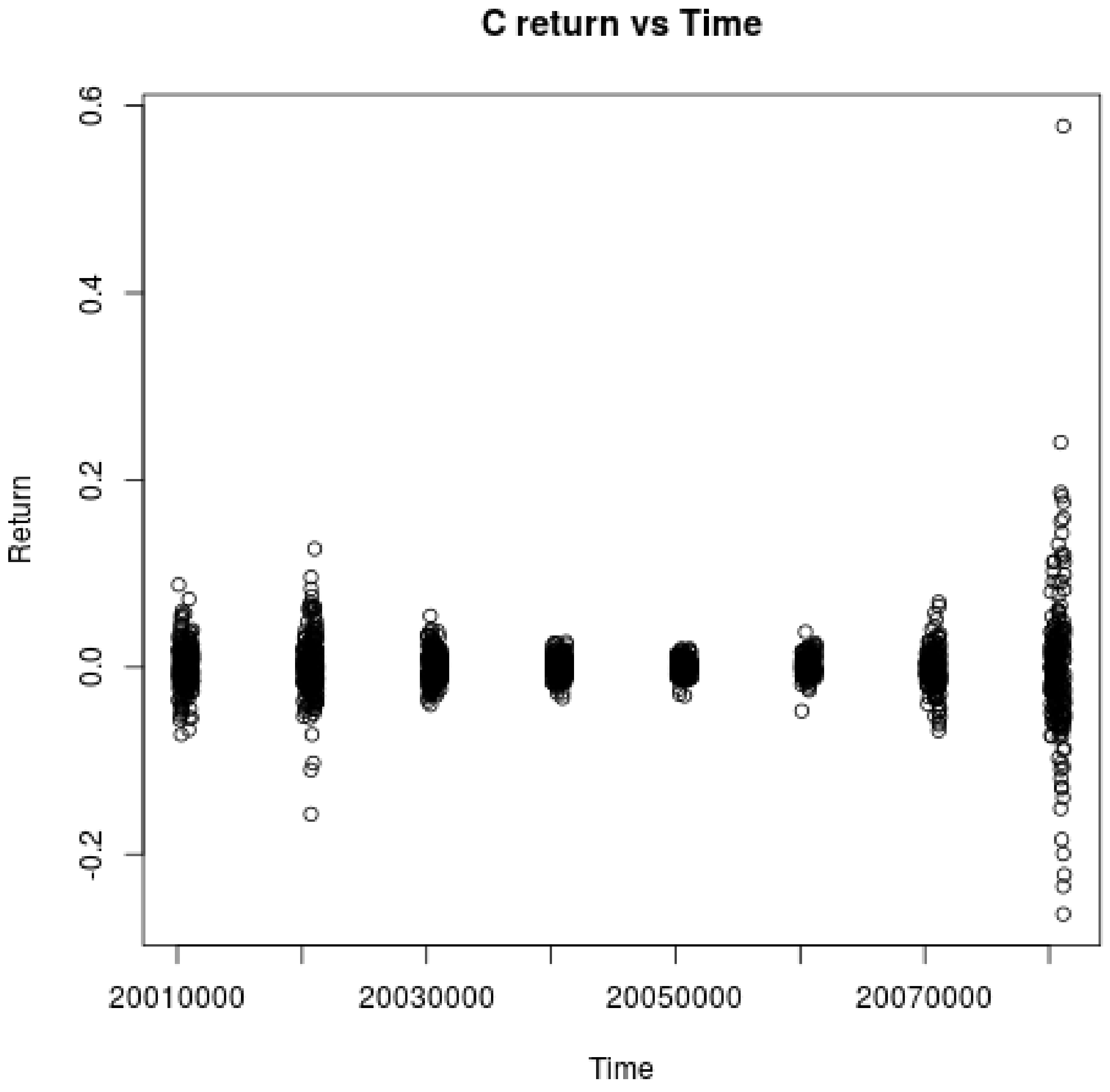
Abheek Ghosh
140123047

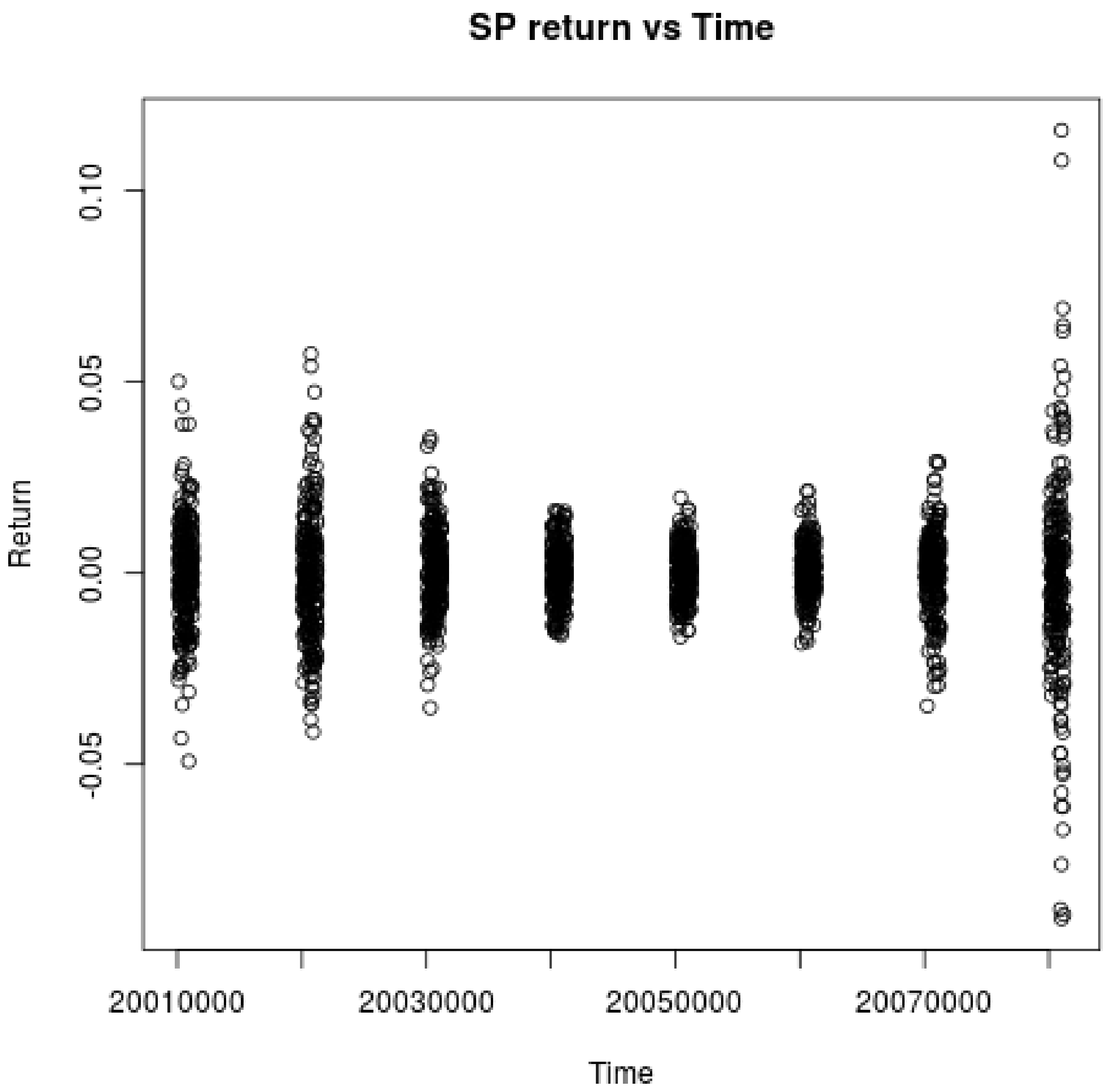
August 14, 2017

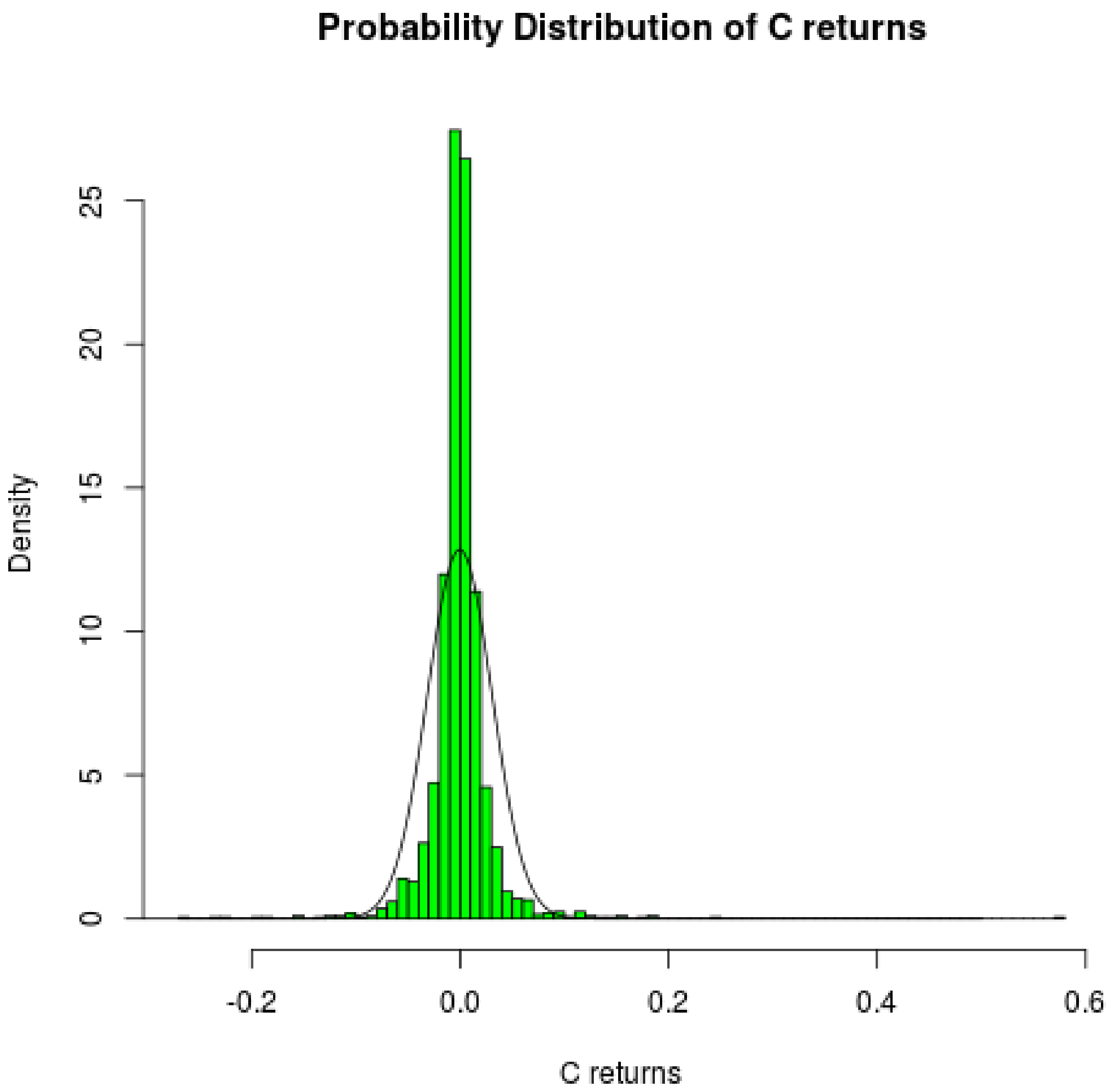
Question 1

Code for R

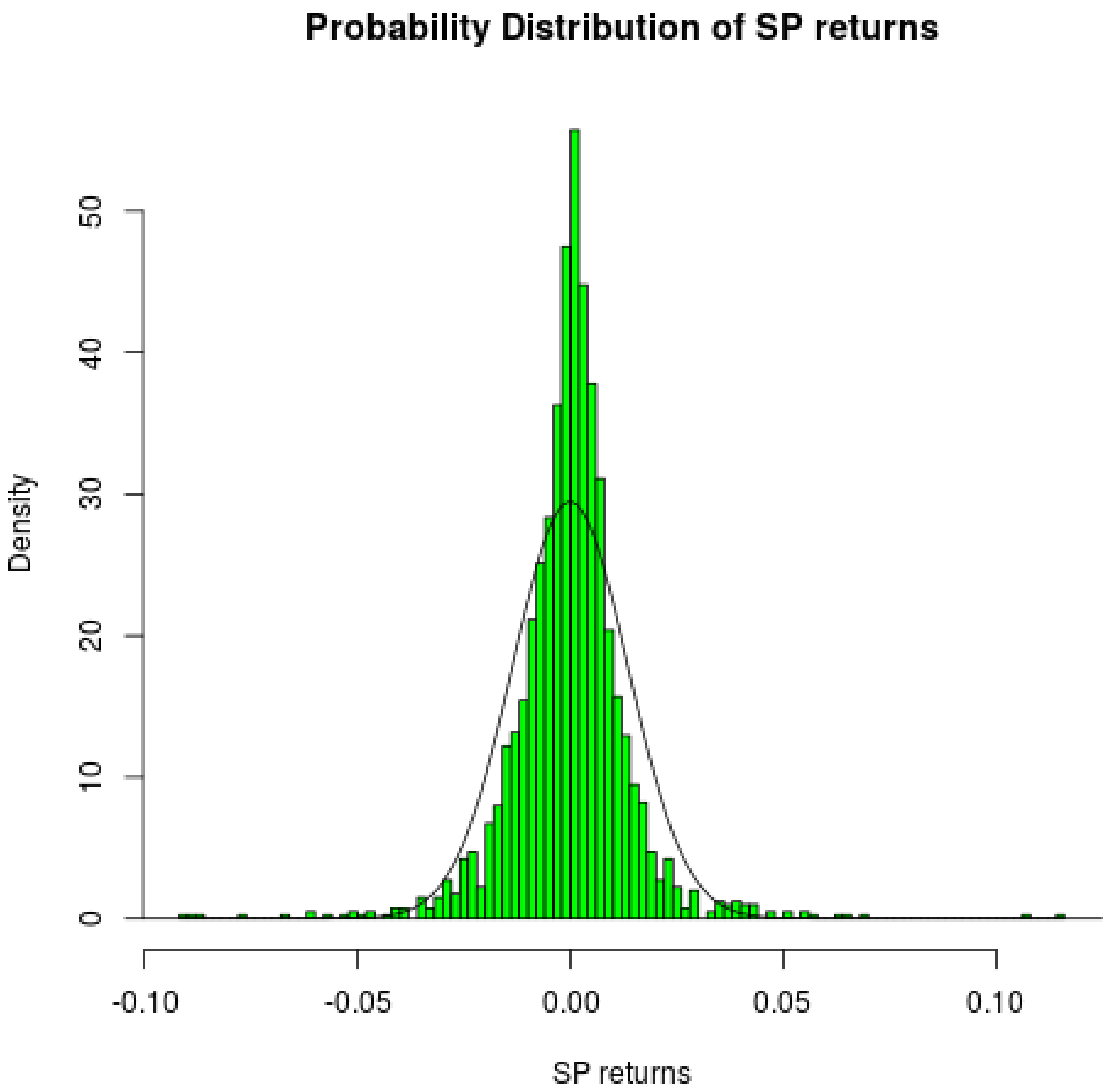
```
rm(list = ls())
d = read.table("d-csp0108.txt", header=TRUE)

k = 2
plot(d[,1], d[,k], main="C return vs Time", xlab="Time", ylab="Return")
dev.copy(png, "plots/plota1.png")
dev.off ()
k = 3
plot(d[,1], d[,k], main="SP return vs Time", xlab="Time", ylab="Return")
dev.copy(png, "plots/plota2.png")
dev.off ()

k = 2
hist(d[,k], freq=FALSE, breaks=100, col='green', main='Probability Distribution of C returns', xlab='C returns')
x <- seq(-0.5, 0.5, length=1000)
lines(x, dnorm(x, mean(d[,k]), sd(d[,k])))
dev.copy(png, "plots/plota3.png")
dev.off ()

k = 3
hist(d[,k], freq=FALSE, breaks=100, col='green', main='Probability Distribution of SP returns', xlab='SP returns')
x <- seq(-0.5, 0.5, length=1000)
lines(x, dnorm(x, mean(d[,k]), sd(d[,k])))
dev.copy(png, "plots/plota4.png")
dev.off ()
```


C return vs Time
Return
0.6
0.4
0.2
0.0
-0.2
20010000
20030000
20050000
20070000
Time


SP return vs Time
Return
0.10
0.05
0.00
-0.05
20010000
20030000
20050000
20070000
Time



Probability Distribution of C returns
Density
25
20
15
10
5
0
-0.2
0.0
0.2
0.4
0.6
C returns

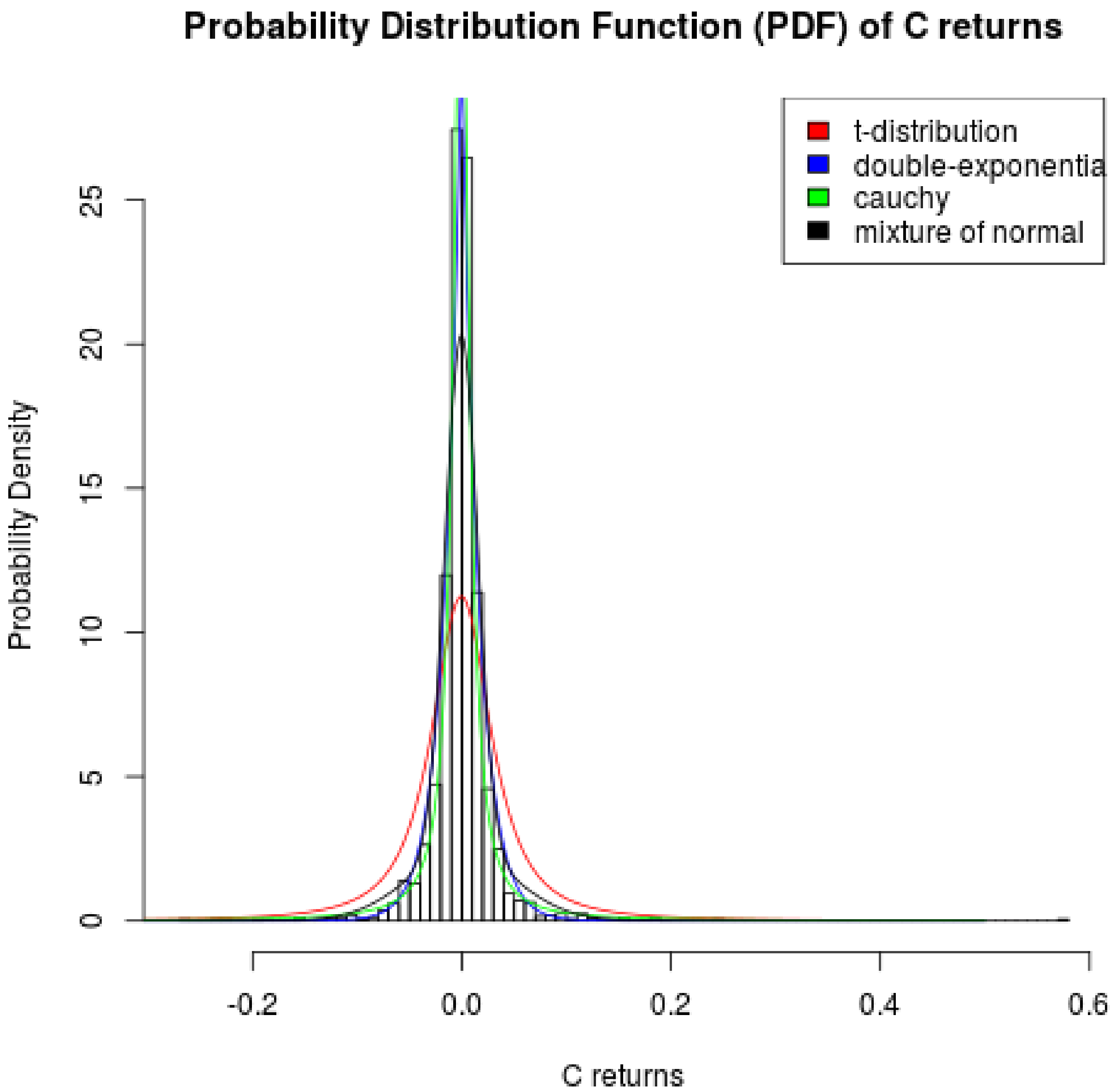

Probability Distribution of SP returns
Density
0
10
20
30
40
50
-0.10
-0.05
0.00
0.05
0.10
SP returns

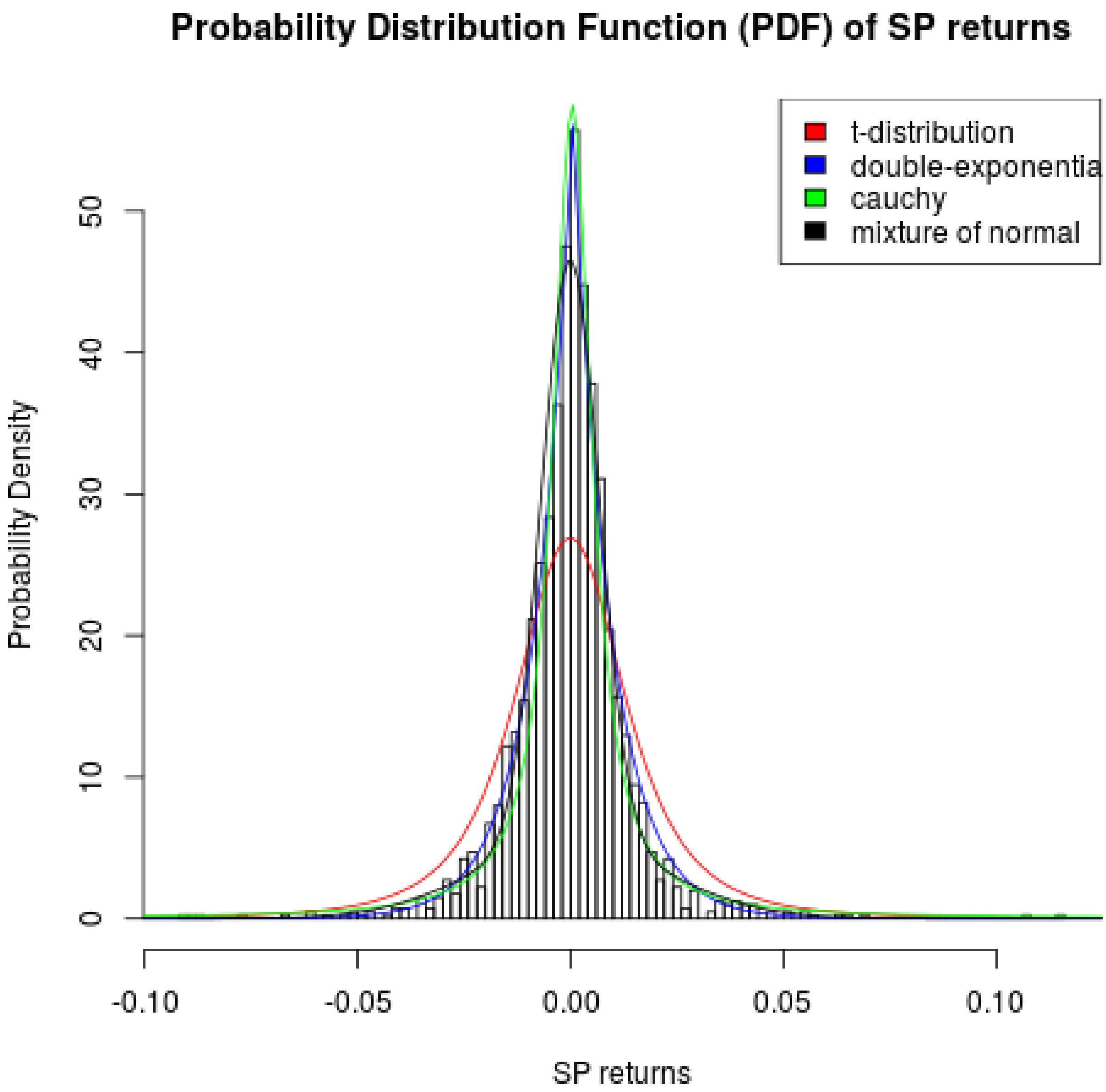
Question 2

Code for R

```r
rm(list = ls())
library(MASS)

ddoublex = function(x, mu = 0, lambda = 1) {
    a = abs(x - mu)
    return (dexp(a, lambda)/2)
}

dmixnorm = function(x, p = 0.5, mu1 = 0, mu2 = 0, sig1 = 1, sig2 = 100) {
    return (p*dnorm(x, mu1, sig1) + (1-p)*dnorm(x, mu2, sig2))
}

d = read.table("d-csp0108.txt", header=TRUE)

names = c('C', 'SP')
cols = c('red', 'blue', 'green', 'black')
dists = c('t-distribution', 'double-exponential', 'cauchy', 'mixture of normal')

for (k in 2:3) {
    x <- seq(-0.5, 0.5, length=1000)
    mu_d = mean(d[,k])
    sd_d = sd(d[,k])

    hist(d[,k], prob=T, breaks=100, main = sprintf("Probability Distribution Function (PDF) of %s returns", names[k-1]),
        xlab = sprintf("%s returns", names[k-1]), ylab = "Probability Density")


    # t-dist
    param = fitdistr(d[,k], "t", start = list(m = mu_d, s = sd_d, df=3))#, lower=c(-1, 0.001,1))
    param = param$estimate
    lines(x, dt((x-mu_d)/sd_d, df=param[3])/sd_d, col = cols[1])

    param = fitdistr(d[,k], ddoublex, start = list(mu = 0, lambda = 1))
    param = param$estimate
    # print(param)
    lines(x, ddoublex((x), mu = param[1], lambda = param[2]),  col = cols[2])

    param = fitdistr(d[,k], "cauchy", start = list(location = 0, scale = 1))
    param = param$estimate
    lines(x, dcauchy((x-mu_d), location = param[1], scale = param[2]),  col = cols[3])

    sd1 = sd_d/2
```

```
    p = 0.7
    param = c(p, mu_d, mu_d, sd1, ( (sd_d^2 - p*sd1^2) / (1-p) )^0.5)
    # print(param)
    lines(x, dmixnorm(x, p = param[1], mu1 = param[2], mu2 = param[3], sig1 = param[4], sig2 =
        param[5]),  col = cols[4])

    legend("topright", legend = dists, fill = cols)
    dev.copy(png, sprintf("plots/plotb%d.png", k-1))
    dev.off ()
}
```




Probability Distribution Function (PDF) of SP returns
t-distribution
double-exponentia
cauchy
mixture of normal
Probability Density
0
10
20
30
40
50
-0.10
-0.05
0.00
0.05
0.10
SP returns

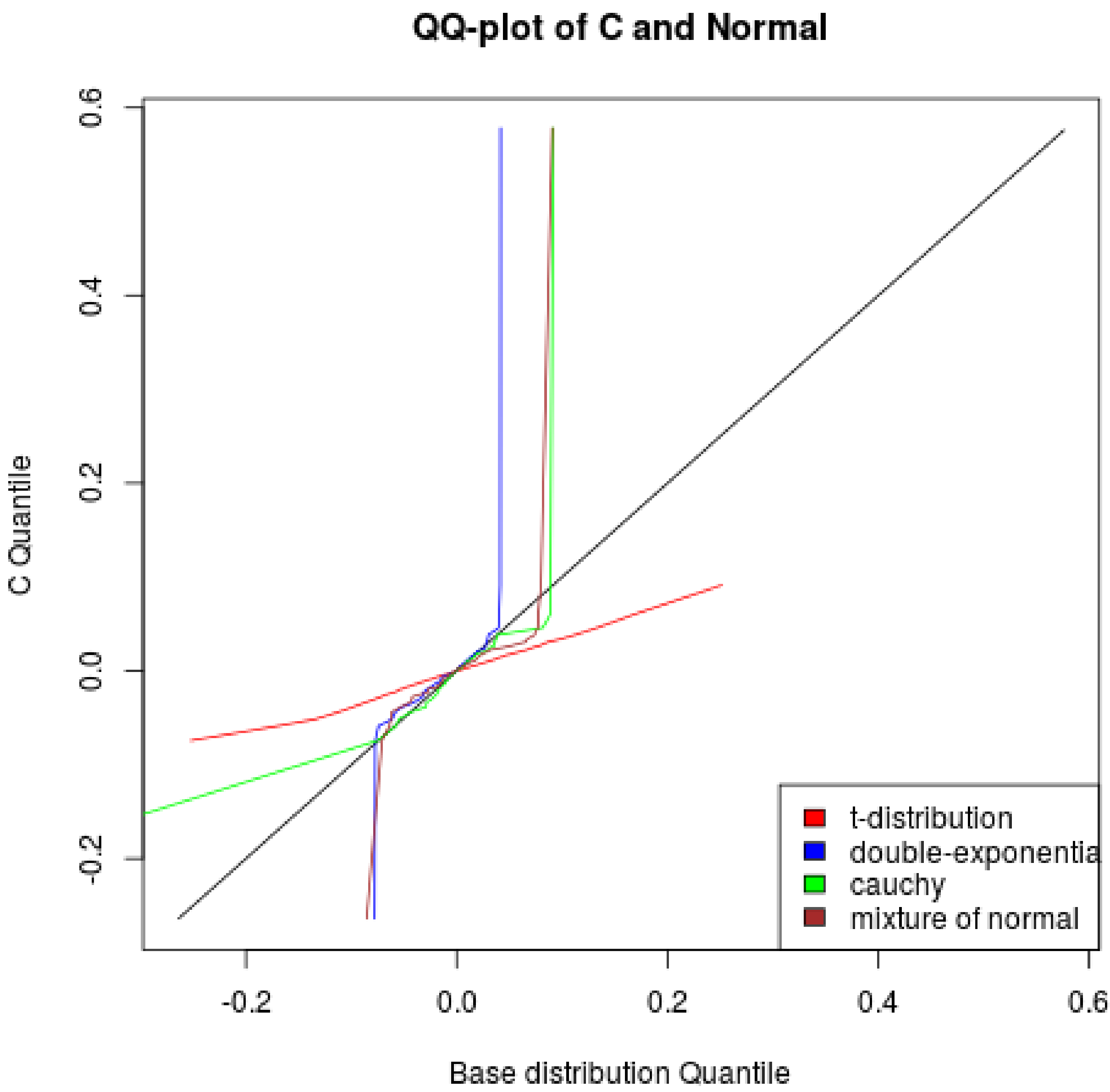
From the plots we observe that the distribution of the data given fits very well with the assumed distribution. The probability distribution function (PDF) and the survival function both fit very well. From the Quantile-Quantile plot we observe that the given data is heavier tailed with respect to double exponential distribution and mixed normal distribution. On the other hand, t-distribution and the cauchy distribution are heavier tailed than the given data.

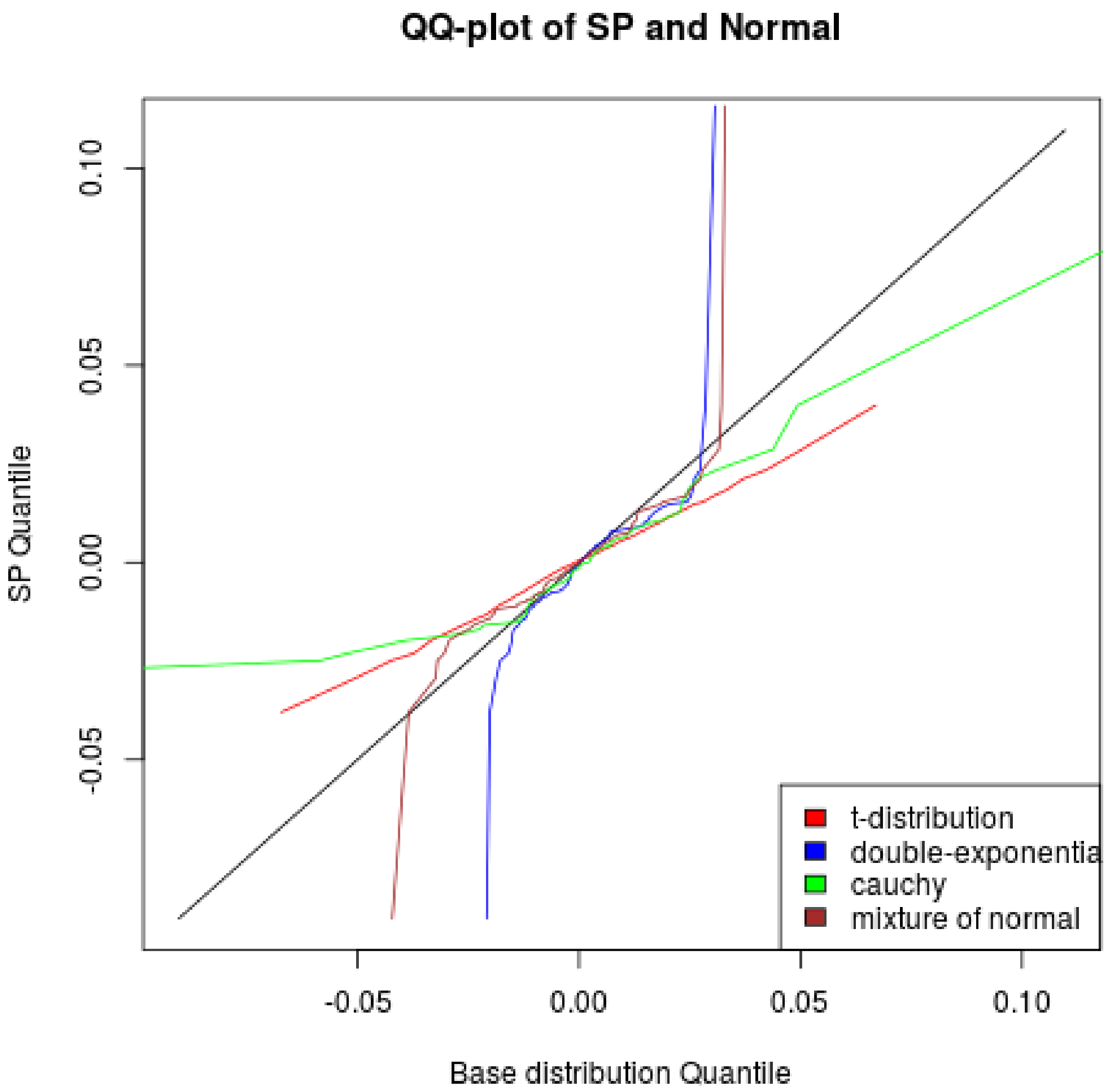
Question 3

Code for R

```
rm(list = ls())
library(MASS)

ddoublex = function(x, mu = 0, lambda = 1) {
    a = abs(x - mu)
    return (dexp(a, lambda)/2)
}

dmixnorm = function(x, p = 0.5, mu1 = 0, mu2 = 0, sig1 = 1, sig2 = 100) {
    return (p*dnorm(x, mu1, sig1) + (1-p)*dnorm(x, mu2, sig2))
}

rdoublex = function(n, mu = 0, lambda = 1) {
    D = rexp(n, lambda)
    temp = runif(n)
    D[temp > 0.5] = -D[temp > 0.5]
    D = D + mu
    return(D)
}

rmixnorm = function(n, p = 0.5, mu1 = 0, mu2 = 0, sig1 = 1, sig2 = 100) {
    n1 = as.integer(n*p)
    D1 = rnorm(n1, mu1, sig1)
    D2 = rnorm(n - n1, mu2, sig2)
    D = c(D1, D2)
    return(D)
}

d = read.table("d-csp0108.txt", header=TRUE)

names = c('C', 'SP')
cols = c('red', 'blue', 'green', 'brown')
dists = c('t-distribution', 'double-exponential', 'cauchy', 'mixture of normal')

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

```r
  Q_data = quantile(d[,k], probs = p)
  mu_d = mean(d[,k])
  sd_d = 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 = "Base distribution Quantile", ylab = sprintf("%s Quantile", names[k-1]), col = 'black', type = 'l')

  # t-dist
  param = fitdistr(d[,k], "t", start = list(m = mu_d, s = sd_d, df=3))#, lower=c(-1, 0.001,1))
  param = param$estimate
  # lines(x, dt((x-mu_d)/sd_d, df=param[3])/sd_d, col = cols[1])
  Q_t = qt(p, df=param[3])*sd_d + mu_d
  lines(Q_t, Q_data, col = cols[1])

  param = fitdistr(d[,k], ddoublex, start = list(mu = 0, lambda = 1))
  param = param$estimate
  # print(param)
  Q_doublex = quantile(rdoublex(length(p), mu = param[1], lambda = param[2]), probs = p)
  lines(Q_doublex, Q_data, col = cols[2])

  param = fitdistr(d[,k], "cauchy", start = list(location = 0, scale = 1))
  param = param$estimate
  Q_cauchy = quantile(rcauchy(length(p), location = param[1], scale = param[2]), probs = p)
  lines(Q_cauchy, Q_data, col = cols[3])

  sd1 = sd_d/2
  pp = 0.7
  param = c(pp, mu_d, mu_d, sd1, ( (sd_d^2 - pp*sd1^2) / (1-pp) )^0.5)
  Q_mixnorm = quantile(rmixnorm(length(p), p = param[1], mu1 = param[2], mu2 = param[3], sig1 = param[4], sig2 = param[5]), probs = p)
  lines(Q_mixnorm, Q_data, col = cols[4])

  legend("bottomright", legend = dists, fill = cols)
  dev.copy(png, sprintf("plots/plotc%d.png", k-1))
  dev.off()
}
```


QQ-plot of C and Normal
C Quantile
Base distribution Quantile
0.6
0.4
0.2
0.0
-0.2
-0.2
0.0
0.2
0.4
0.6
t-distribution
double-exponentia
cauchy
mixture of normal


QQ-plot of SP and Normal
SP Quantile
-0.05
0.00
0.05
0.10
Base distribution Quantile
-0.05
0.00
0.05
0.10
t-distribution
double-exponentia
cauchy
mixture of normal

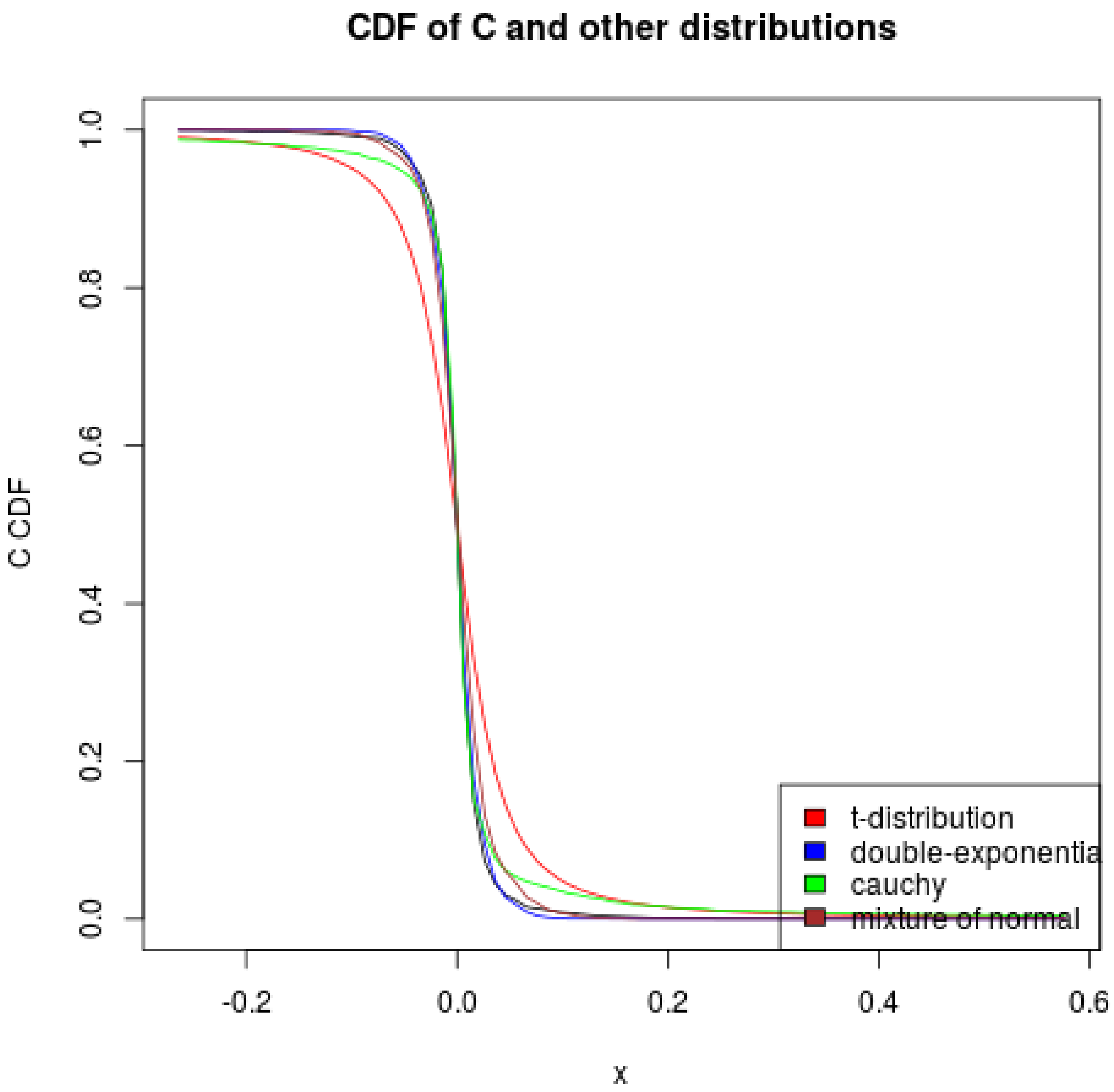
From the plots we observe that the distribution of the data given fits very well with the assumed distribution. From the Quantile-Quantile plot we observe that the given data is heavier tailed with respect to double exponential distribution and mixed normal distribution. On the other hand, t-distribution and the cauchy distribution are heavier tailed than the given data.

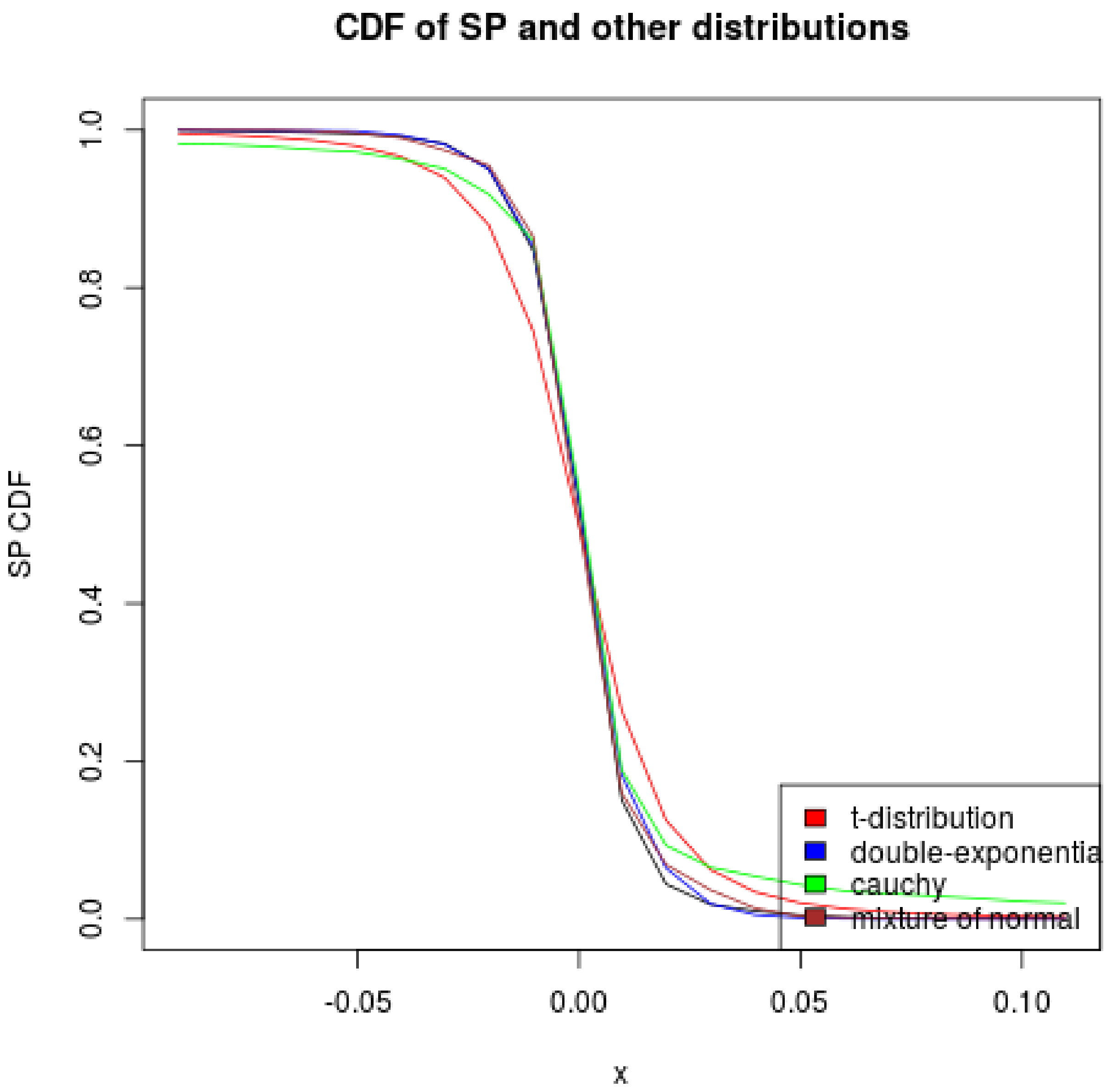
Question 4

Code for R

```
rm(list = ls())
library(MASS)

ddoublex = function(x, mu = 0, lambda = 1) {
    a = abs(x - mu)
    return (dexp(a, lambda)/2)
}

dmixnorm = function(x, p = 0.5, mu1 = 0, mu2 = 0, sig1 = 1, sig2 = 100) {
    return (p*dnorm(x, mu1, sig1) + (1-p)*dnorm(x, mu2, sig2))
}

rdoublex = function(n, mu = 0, lambda = 1) {
    D = rexp(n, lambda)
    temp = runif(n)
    D[temp > 0.5] = -D[temp > 0.5]
    D = D + mu
    return(D)
}

rmixnorm = function(n, p = 0.5, mu1 = 0, mu2 = 0, sig1 = 1, sig2 = 100) {
    n1 = as.integer(n*p)
    D1 = rnorm(n1, mu1, sig1)
    D2 = rnorm(n - n1, mu2, sig2)
    D = c(D1, D2)
    return(D)
}

d = read.table("d-csp0108.txt", header=TRUE)

names = c('C', 'SP')
cols = c('red', 'blue', 'green', 'brown')
dists = c('t-distribution', 'double-exponential', 'cauchy', 'mixture of normal')

for (k in 2:3) {
    x = seq(min(d[,k]), max(d[,k]), 0.01)
    mu_d = mean(d[,k])
```

```r
  sd_d = sd(d[,k])
  cdf_d = ecdf(d[,k])

  plot(x, 1- cdf_d(x), main = sprintf("CDF of %s and other distributions", names[k-1]),
     xlab = "x", ylab = sprintf("%s CDF", names[k-1]), col = 'black', type = 'l')

  # t-dist
  param = fitdistr(d[,k], "t", start = list(m = mu_d, s = sd_d, df=3))#, lower=c(-1, 0.001,1))
  param = param$estimate
  lines(x, 1-pt((x-mu_d)/sd_d, df=param[3]) , col = cols[1])

  param = fitdistr(d[,k], ddoublex, start = list(mu = 0, lambda = 1))
  param = param$estimate
  cdf_base = ecdf(rdoublex(length(d[,k]), mu = param[1], lambda = param[2]))
  lines(x, 1-cdf_base(x) , col = cols[2])

  param = fitdistr(d[,k], "cauchy", start = list(location = 0, scale = 1))
  param = param$estimate
  cdf_base = ecdf(rcauchy(length(d[,k]), location = param[1], scale = param[2]))
  lines(x, 1-cdf_base(x) , col = cols[3])

  sd1 = sd_d/2
  pp = 0.7
  param = c(pp, mu_d, mu_d, sd1, ( (sd_d^2 - pp*sd1^2) / (1-pp) )^0.5)
  cdf_base = ecdf(rmixnorm(length(d[,k]), p = param[1], mu1 = param[2], mu2 = param[3], sig1
     = param[4], sig2 = param[5]))
  lines(x, 1-cdf_base(x) , col = cols[4])

  legend("bottomright", legend = dists, fill = cols)
  dev.copy(png, sprintf("plots/plotd%d.png", k-1))
  dev.off ()
}
```


CDF of C and other distributions
C CDF
x
t-distribution
double-exponentia
cauchy
mixture of normal
0.0
0.2
0.4
0.6
0.8
1.0
-0.2
0.0
0.2
0.4
0.6


CDF of SP and other distributions
SP CDF
0.0
0.2
0.4
0.6
0.8
1.0
-0.05
0.00
0.05
0.10
x
t-distribution
double-exponentia
cauchy
mixture of normal

From the plots we observe that the distribution of the data given fits very well with the assumed distribution. The probability distribution function (PDF) and the survival function both fit very well. From the Quantile-Quantile plot we observe that the given data is heavier tailed with respect to double exponential distribution and mixed normal distribution. On the other hand, t-distribution and the cauchy distribution are heavier tailed than the given data.