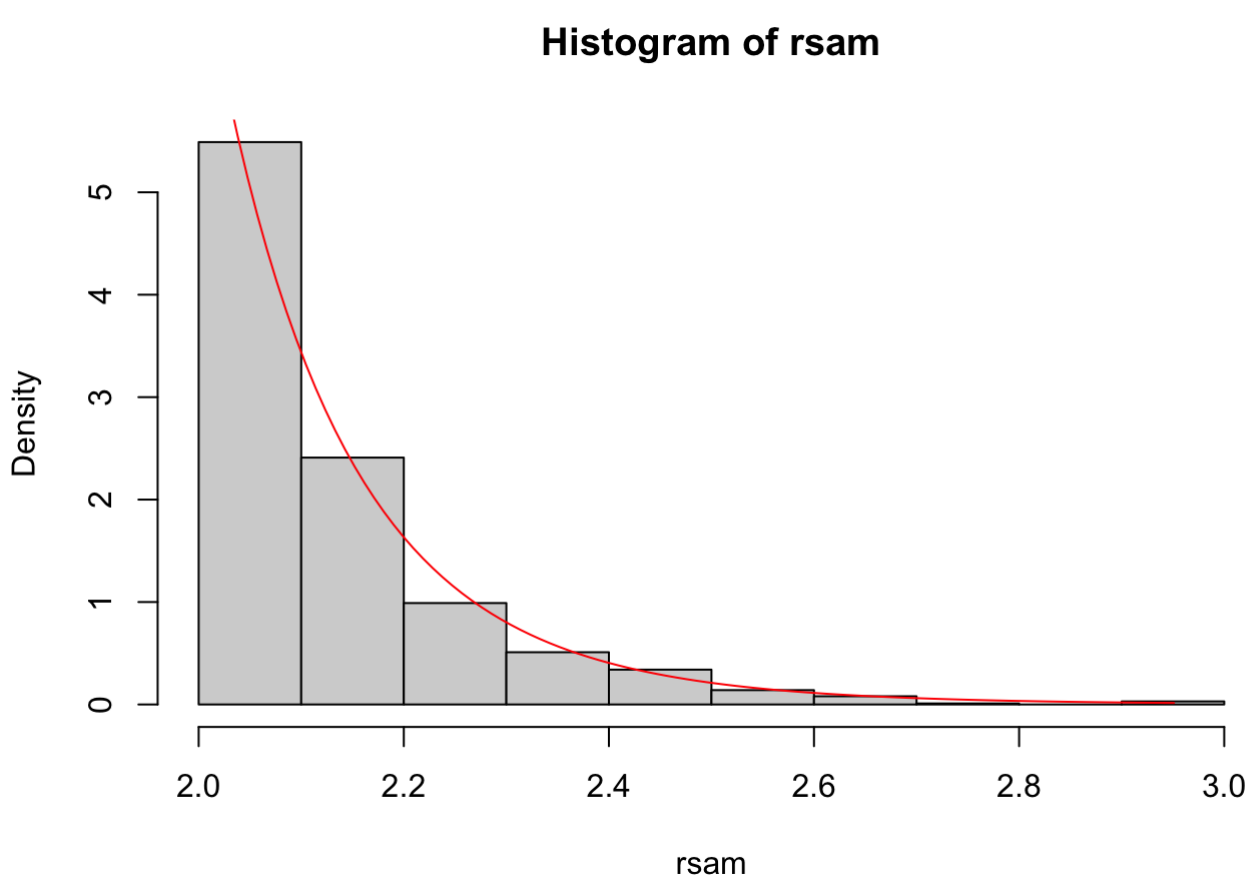
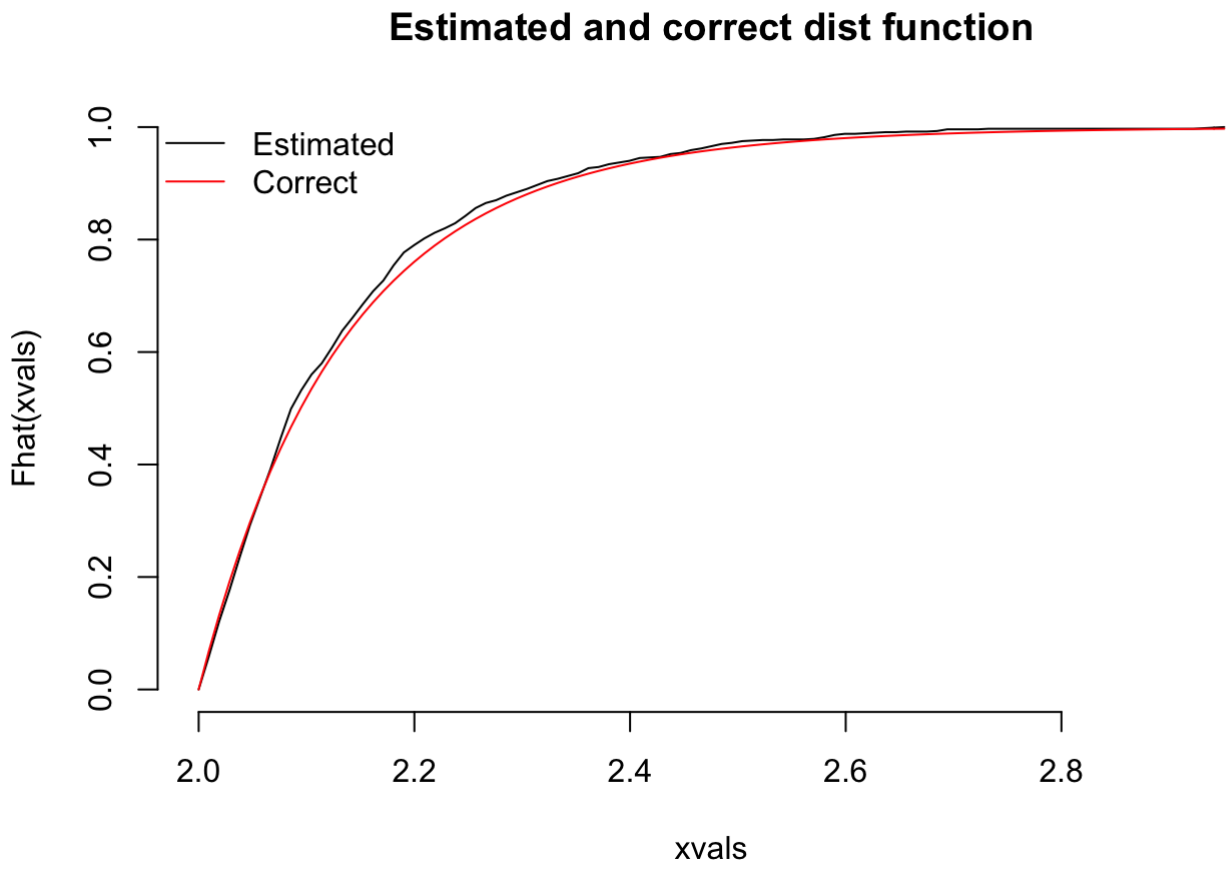
# Estimating Distribution of Pareto

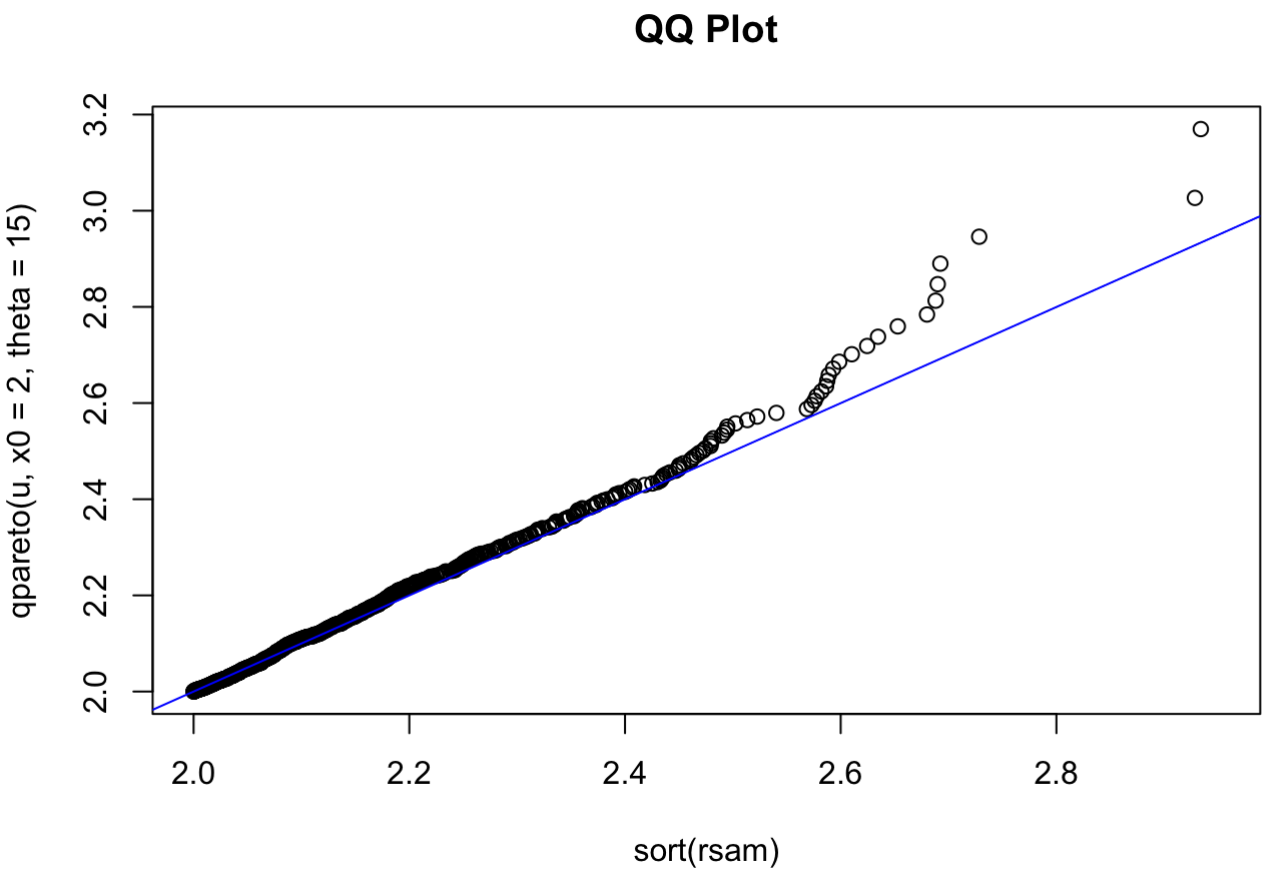
```
qpareto <- function(u, x0, theta){
# checks if values are valid
stopifnot(all(x0 > 0), all(theta > 0), all(u >= 0), all(u <= 1))
# below is the quantile function formula
# which generates random paretos as proven in previous question
x0 * (1 - u)^(-1/theta)
}

rpareto <- function(n, x0, theta){
# checks if values are valid
stopifnot(all(x0 > 0), all(theta > 0), n >= 0)
# u is a vector containing n uniform random numbers from (0,1)
u <- runif(n)
qpareto(u, x0 = x0, theta = theta)
}


# returns the density of a pareto
dpareto <- function(x, x0, theta){
stopifnot(all(x0 > 0), all(theta > 0))
ifelse(x < x0, 0, (theta / x) * (x0 / x)^theta)
}

# true pmf of pareto
ppareto <- function(x, x0, theta){
stopifnot(all(x0 > 0), all(theta > 0))
ifelse(x < x0, 0, 1 - (x0 / x)^theta)
}



rsam <- rpareto(1000, x0=2,theta=15)

# creates histogram with y axis representing density
hist(rsam, freq=FALSE)
# creates a sequence of values from 2 to rsam maximum with 101 values in this sequenc
e
xvals <- seq(from=2, to=max(rsam), length=101)
# draws a red line of our values against their density
lines(xvals, dpareto(xvals, x0=2, theta=15), col='red')
```




```
# estimate of distribution function in R
Fhat <- function(x) {
  # creating a vector where each item is the proportion of that item seen in rsam
  sapply(x, function(xv) mean(rsam <= xv))
}


plot(xvals, Fhat(xvals), type = 'l', main='Estimated and correct dist function', bty=
'n')
lines(xvals, ppareto(xvals, x0=2, theta=15), col='red')
legend('topleft', legend=c('Estimated', 'Correct'), lty=1, pch=NA, col=c('black', 're
d'), bty='n')
```




```
# estimate of quantiles to true value
n <- length(rsam)
u <- (1:n) / n
plot(sort(rsam), qpareto(u, x0=2, theta=15), main='QQ Plot')
abline(a = 0, b=1, col='blue')
```


QQ Plot
qpareto(u, x0 = 2, theta = 15)
2.0
2.2
2.4
2.6
2.8
3.0
3.2
sort(rsam)
2.0
2.2
2.4
2.6
2.8