In [1]:

```
library(MASS)
data(quine)
```

In [2]:

```
p.hat = function(k, r=1.5) {
    return(sum(k) / (length(k) * r + sum(k)))
}
```

Q1a) Answer:

$\hat{p}$ = 0.916475972540046

In [3]:

```
p.post = function(k, r=1.5, a=0.5, b=0.5) {
    return (c(sum(k) + a, length(k) * r + b))
}
```

In [4]:

```
p.post(quine$Days)
```

2403.5 219.5

Q1b) Answer:

α = 2403.5, β = 219.5

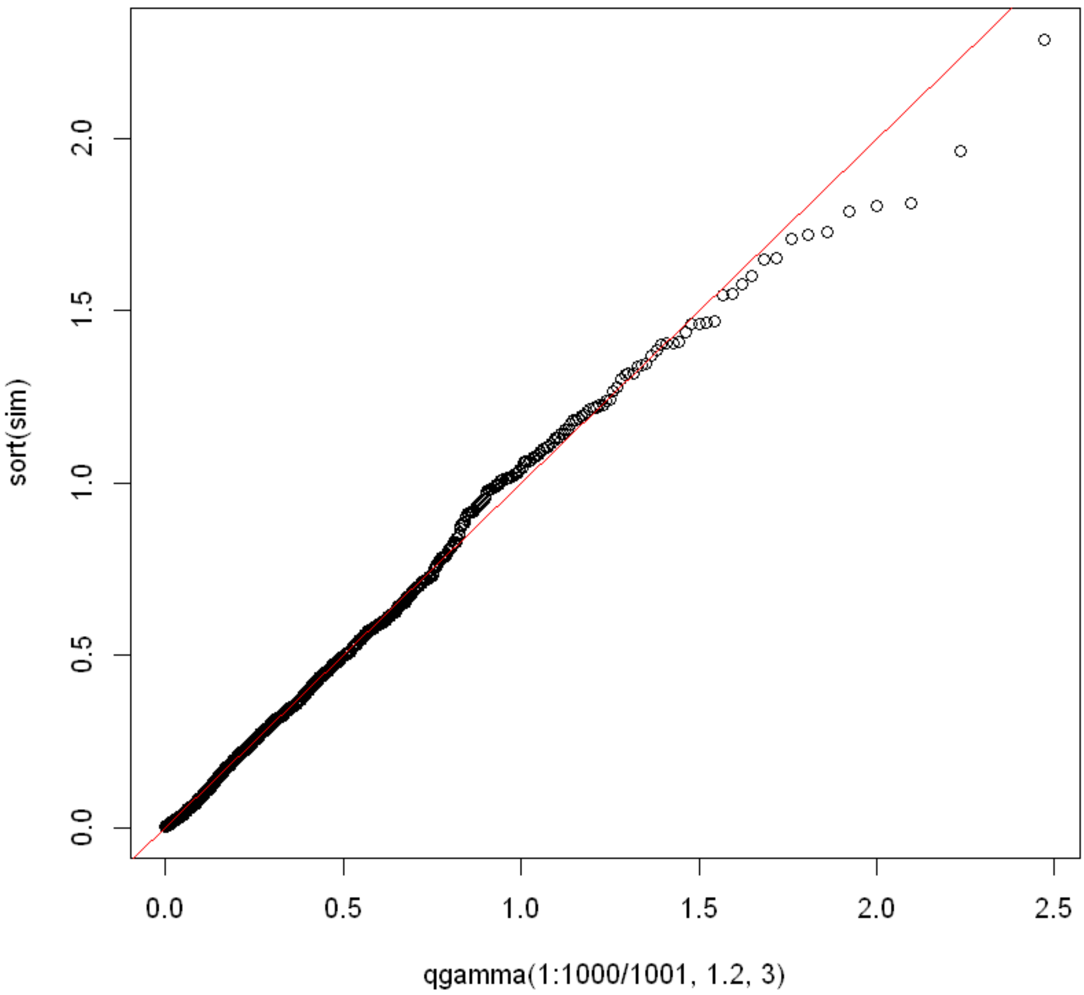
In [16]:

```
h1 = function(x, c, d) {
    return (d * (1 + c * x)^3)
}

g = function(x, c, d) {
    return (d * log((1 + c * x) ^ 3) - d * (1 + c * x)^3 + d)
}

f1 = function(x, alpha) {
    d = alpha - 1/3
    c = 1 / sqrt(9 * d)

    h1.prime = function(x, c, d) {
        return (3 * d * c * (1 + c * x)^2)
    }

    return (h1(x, c, d)^(alpha - 1) * exp(-h1(x, c, d)) * h1.prime(x, c, d))
}

f2 = function(x, alpha) {
    d = alpha - 1/3
    c = 1 / sqrt(9 * d)
    return (exp(g(x, c, d)))
}

cmp = function(alpha) {
    return (f1(1, alpha) / exp(f2(1, alpha)))
}

h2 = function(x) {
    return (exp(-x^2 / 2))
}

gamma = function(alpha = 1, beta = 1) {
    d = alpha - 1/3;
    c = 1 / sqrt(9 * d)
    r = cmp(alpha)

    while (TRUE) {
        x = rnorm(1)
        y = runif(1)

        if (x > -1/c && y < f1(x, alpha) / h2(x) * r) {
            break
        }
    }

    return (h1(x, c, d) / beta)
}

rgamma_ = function(n_iter, alpha = 1, beta = 1) {
    return (replicate(n_iter, gamma(alpha, beta)))
}
```

Q2c) Answer:

In [18]:

```
sim = rgamma_(1000, 1.2, 3)
plot(qgamma(1:1000/1001, 1.2, 3), sort(sim))
abline(0, 1, col="red")
```



In [ ]: