

R for Linear Stat (I)

Pi-Wen Tsai

September 22, 2008

1 Some Basic R

Numeric vector

```
> options(keep.source = TRUE, width = 60)
```

```
> a = c(2, 4, 6, 7, 8)
```

```
> a[1]
```

```
[1] 2
```

```
> a[c(1, 4, 5)]
```

```
[1] 2 7 8
```

```
> a[c(1:4)]
```

```
[1] 2 4 6 7
```

```
> a < 5
```

```
[1] TRUE TRUE FALSE FALSE FALSE
```

```
> a[a < 5]
```

```
[1] 2 4
```

```
> seq(from = 1, to = 10, by = 0.5)
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
```

```
[12] 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0
```

```
> seq(1, 10, 0.5)
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
```

```
[12] 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0
```

```
> rep(1, 4)
```

```
[1] 1 1 1 1
```

```

> b = a
> a + b

[1] 4 8 12 14 16

> a - b

[1] 0 0 0 0 0

> a * b

[1] 4 16 36 49 64

> x = rbind(a, b)
> x

  [,1] [,2] [,3] [,4] [,5]
a    2    4    6    7    8
b    2    4    6    7    8

> y = cbind(a, b)
> y

      a b
[1,] 2 2
[2,] 4 4
[3,] 6 6
[4,] 7 7
[5,] 8 8

      Matrix

> dim(x)

[1] 2 5

> dim(y)

[1] 5 2

> z = matrix(c(1:10), nrow = 5, ncol = 2)
> z

  [,1] [,2]
[1,]  1    6
[2,]  2    7
[3,]  3    8
[4,]  4    9
[5,]  5   10

```

```
> z = matrix(c(1:10), nrow = 5, ncol = 2, byrow = T)
> z
```

```
      [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
[5,]    9   10
```

```
> z[3, ]
```

```
[1] 5 6
```

```
> z[, 1]
```

```
[1] 1 3 5 7 9
```

```
> y + z
```

```
      a  b
[1,]  3  4
[2,]  7  8
[3,] 11 12
[4,] 14 15
[5,] 17 18
```

```
> x %% z
```

```
      [,1] [,2]
a  165  192
b  165  192
```

```
> x = matrix(c(1:16), 4, 4)
```

```
> x
```

```
      [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
```

```
> diag(x)
```

```
[1]  1  6 11 16
```

```
> eigen(x)
```

```
$values
[1] 3.620937e+01 -2.209373e+00 1.477980e-15 1.332542e-16
```

```
$vectors
      [,1]      [,2]      [,3]      [,4]
[1,] 0.4140028 -0.82289268 -0.2915006 0.1787978
[2,] 0.4688206 -0.42193991 0.7342968 0.1474867
[3,] 0.5236384 -0.02098714 -0.5940919 -0.8313669
[4,] 0.5784562 0.37996563 0.1512957 0.5050823
```

Factors

```
> gender = c(rep("F", 3), rep("M", 4))
> gender = factor(gender)
> levels(gender)
```

```
[1] "F" "M"
```

Data set in R

```
> data(cars)
> names(cars)
```

```
[1] "speed" "dist"
```

```
> y = cars$dist
> x = cars$speed
> mean(y)
```

```
[1] 42.98
```

```
> sum(y)
```

```
[1] 2149
```

```
> median(y)
```

```
[1] 36
```

```
> var(y)
```

```
[1] 664.0608
```

```
> length(y)
```

```
[1] 50
```

```
> max(y)
```

```
[1] 120
```

```
> min(y)
```

```
[1] 2
```

```
> summary(y)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2.00	26.00	36.00	42.98	56.00	120.00

```
> quantile(y, type = 6)
```

0%	25%	50%	75%	100%
2.0	25.5	36.0	57.0	120.0

Some plots

```
> op = par(mfrow = c(2, 2))
```

```
> boxplot(y)
```

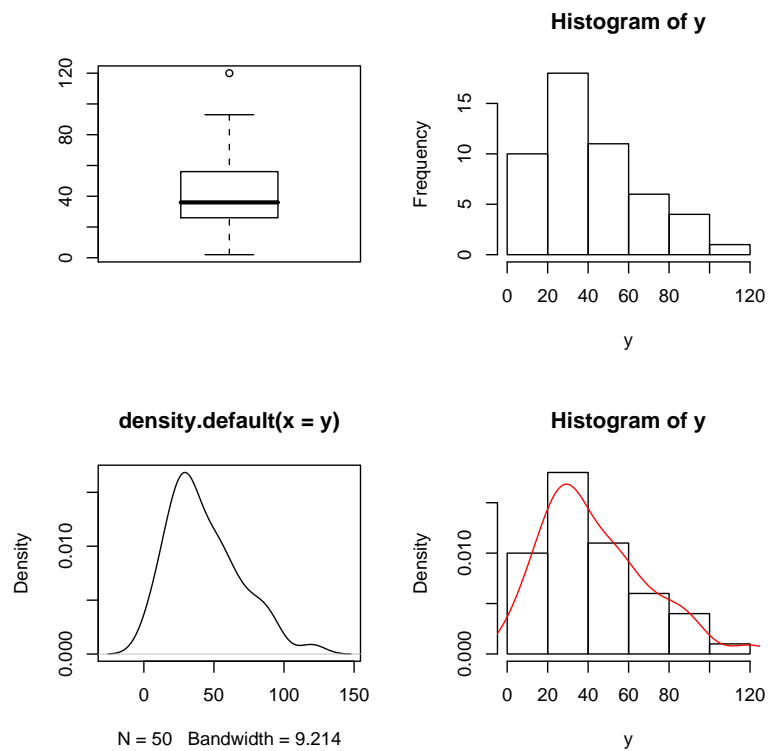
```
> hist(y)
```

```
> plot(density(y))
```

```
> hist(y, prob = T)
```

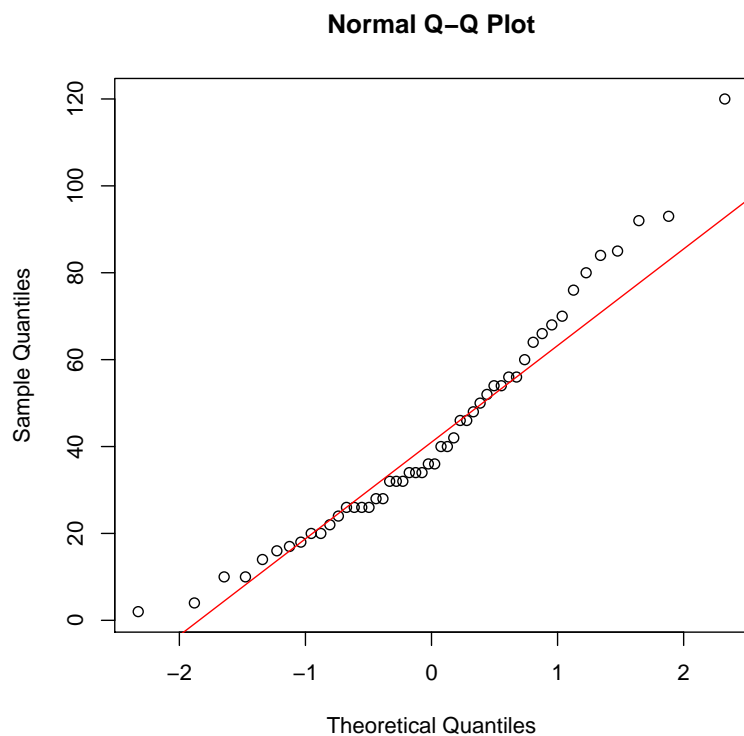
```
> lines(density(y), col = 2)
```

```
> par(op)
```



```
> qqnorm(y)
```

```
> qqline(y, col = 2)
```



Write your own function

```
> std = function(x) {
+   sqrt(x)
+ }
> std
```

```
function (x)
{
  sqrt(x)
}
```

```
> std(4)
```

```
[1] 2
```

draw a random sample from S

```
> options(width = 80)
> S = seq(from = 1, to = 30, by = 2)
> sample(S, 8)
```

```
[1] 9 23 29 3 7 11 15 17
```

```
> sample(S, 10, replace = TRUE)
```

```
[1] 13 15 23 25 7 1 15 29 5 1
```

2 Some Basic R programs for probability distributions

```
> pnorm(0)

[1] 0.5

> qnorm(0.5)

[1] 0

> qnorm(0.95)

[1] 1.644854

> pnorm(4, mean = 3, sd = 1)

[1] 0.8413447

> dnorm(0.5, 3, 1)

[1] 0.0175283

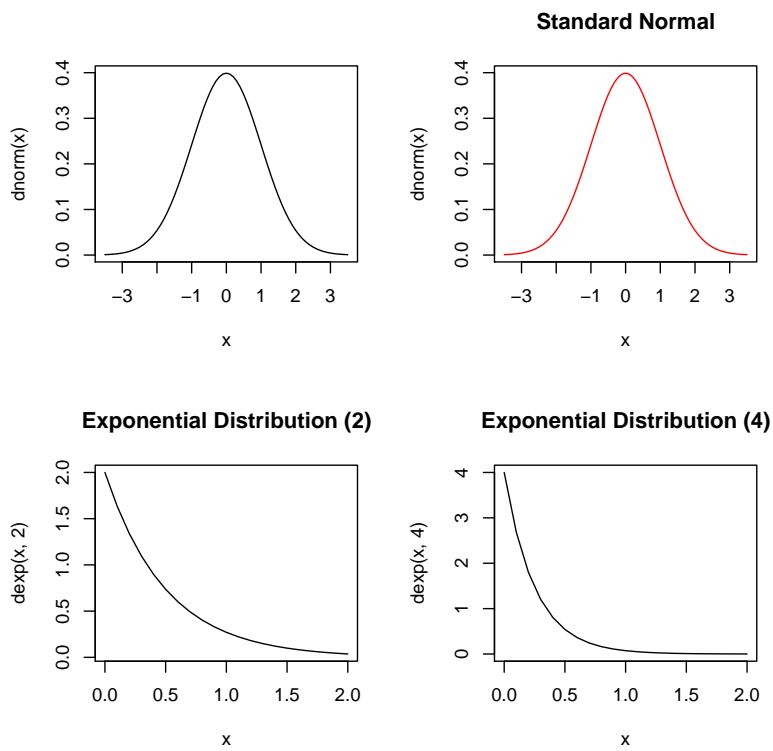
> qnorm(0.95, 3, 1)

[1] 4.644854

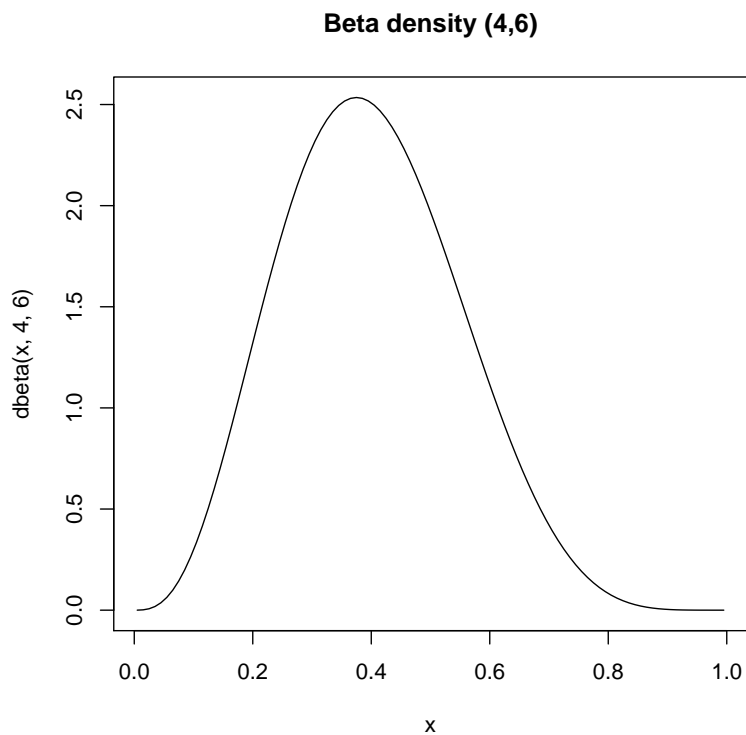
> rnorm(20)

[1] -0.02490469  1.28040741  0.82826324  0.53011686 -1.19690231  0.61841003
[7] -0.15334026 -0.72528119  1.58845855  1.16770078  1.17816379  0.86226713
[13]  0.63476529 -1.05770951 -0.61910471  1.76301954  1.11015438  1.93618295
[19] -0.56463837 -0.34558276

> par(mfrow = c(2, 2))
> x = seq(from = -3.5, to = 3.5, by = 0.1)
> plot(x, dnorm(x), type = "l")
> plot(x, dnorm(x), type = "l", col = 2, main = "Standard Normal")
> x = seq(0, 2, 0.1)
> plot(x, dexp(x, 2), type = "l", main = "Exponential Distribution (2)")
> plot(x, dexp(x, 4), type = "l", main = "Exponential Distribution (4)")
> par(op)
```



```
> x <- seq(0.005, 0.995, by = 0.01)
> plot(x, dbeta(x, 4, 6), type = "l", main = "Beta density (4,6)")
```



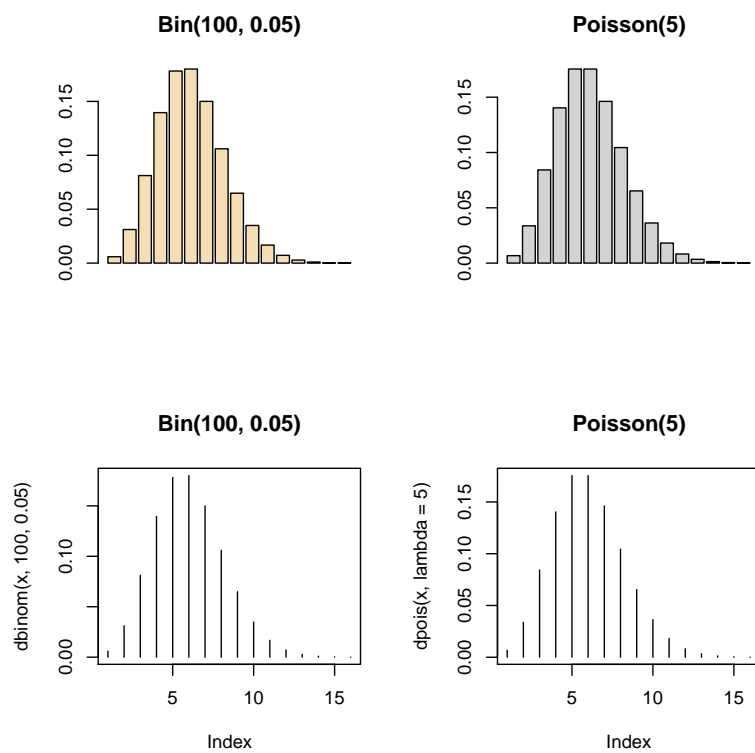
Discrete distribution


```

> dbinom(8, 16, 0.75)
> pbinom(8, 16, 0.75)
> qbinom(0.95, 16, 0.75)
> rbinom(100, 16, 0.75)
> dpois(1, 1.3)

> par(mfrow = c(2, 2))
> x = seq(0, 15, by = 1)
> barplot(dbinom(x, 100, 0.05), col = "wheat", main = "Bin(100, 0.05)")
> barplot(dpois(x, 5), col = "lightgray", main = "Poisson(5)")
> plot(dbinom(x, 100, 0.05), type = "h", main = "Bin(100, 0.05)")
> plot(dpois(x, lambda = 5), type = "h", main = "Poisson(5)")

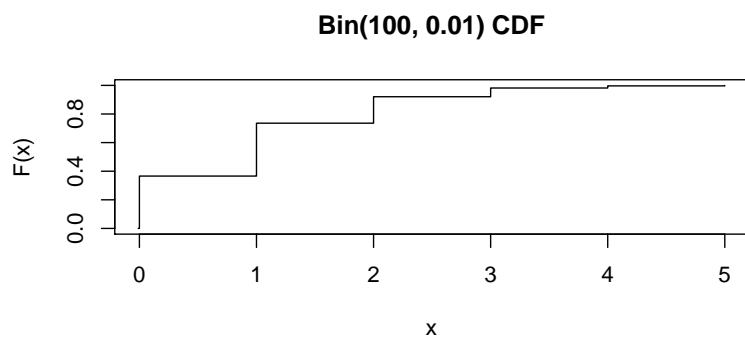
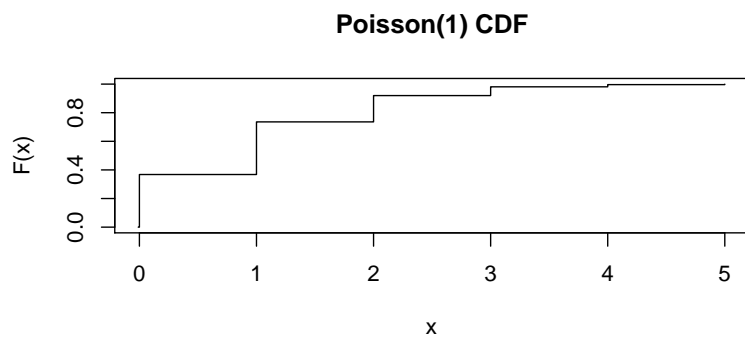
```



```

> options(width = 60)
> par(mfrow = c(2, 1))
> x <- seq(-0.01, 5, 0.01)
> plot(x, ppois(x, 1), type = "s", ylab = "F(x)",
+      main = "Poisson(1) CDF")
> plot(x, pbinom(x, 100, 0.01), type = "s", ylab = "F(x)",
+      main = "Bin(100, 0.01) CDF")
> par(op)

```



Random sample from a given distribution

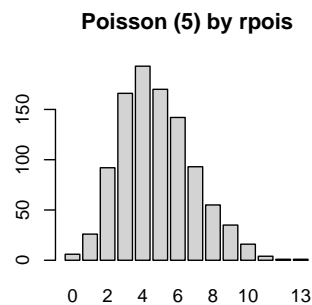
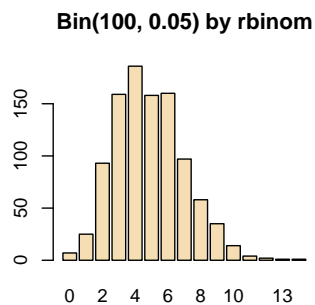
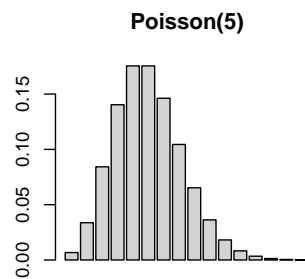
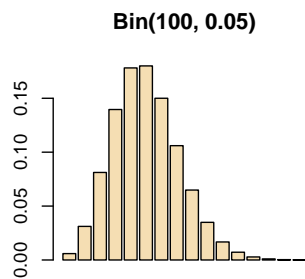
```
> par(mfrow = c(2, 2))
> x = seq(0, 15, by = 1)
> barplot(dbinom(x, 100, 0.05), col = "wheat", main = "Bin(100, 0.05)")
> barplot(dpois(x, 5), col = "lightgray", main = "Poisson(5)")
> Ni = rbinom(1000, 100, p = 0.05)
> table(Ni)
```

```
Ni
 0  1  2  3  4  5  6  7  8  9 10 11 12
10 31 81 140 199 176 152 110 50 31 12 5 3
```

```
> barplot(table(Ni), col = "wheat", main = "Bin(100, 0.05) by rbinom")
> Ni = rpois(1000, 5)
> table(Ni)
```

```
Ni
 0  1  2  3  4  5  6  7  8  9 10 11 12 13
 6 30 78 131 199 185 151 101 57 31 20 5 4 2
```

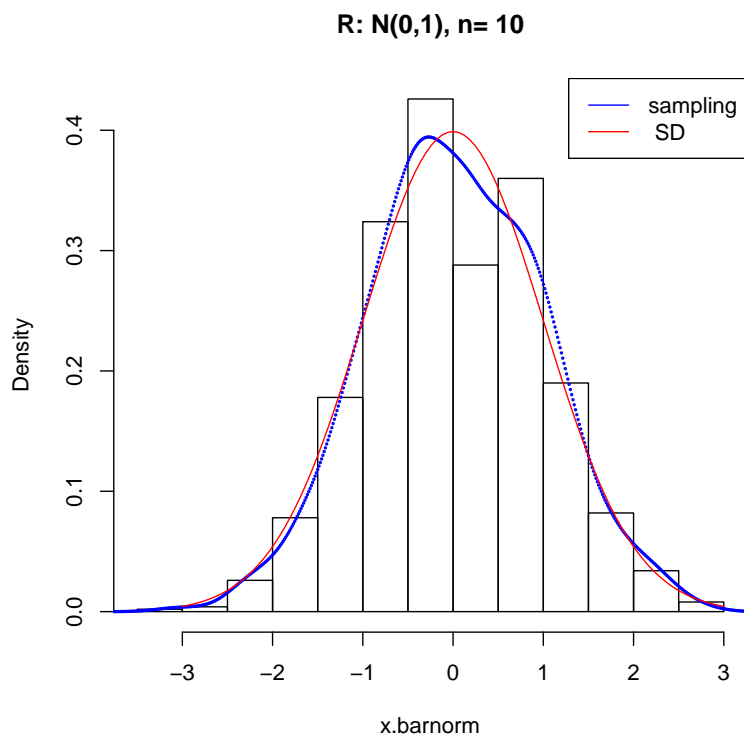
```
> barplot(table(Ni), col = "lightgray", main = "Poisson (5) by rpois")
> par(op)
```



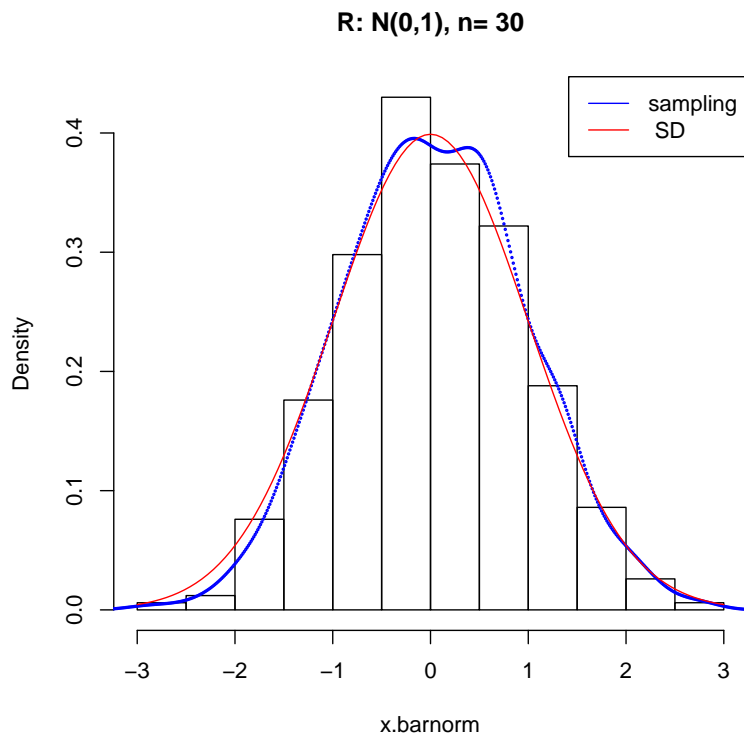
function for central limit theorem a : original population n : sampling size flag: with or without replacement N: repeated sampling

```
> central = function(a, N, n, ind) {
+   mu.a = mean(a)
+   sd.a = sd(a)
+   if (ind == 1)
+     flag = TRUE
+   if (ind == 0)
+     flag = FALSE
+   x.bar = rep(NA, N)
+   for (i in 1:N) x.bar[i] = mean(sample(a, n,
+     replace = flag))
+   x.barnorm = (x.bar - mu.a)/sd.a * sqrt(n)
+   hist(x.barnorm, freq = FALSE, main = paste("R: N(0,1), n=",
+     n))
+   points(density(x.barnorm), cex = 0.2, col = "blue")
+   x = seq(-3, 3, len = 101)
+   y = (1/sqrt(2 * pi)) * exp(-x^2/2)
+   points(x, y, type = "l", xaxt = "n", col = "red")
+   legend("topright", c("sampling", " SD"), lty = 1,
+     col = c("blue", "red"))
+ }

> central(c(1:10), 1000, 10, T)
```



```
> central(c(1:10), 1000, 30, T)
```



```
> duration <- c(5, 10, 6, 11, 5, 14, 30, 11, 17,
+ 3, 9, 3, 8, 8, 5, 5, 7, 4, 3, 7, 9, 11, 11,
```

```

+     9, 4)
> mean(duration)

[1] 8.6

> var(duration)

[1] 32.66667

> summary(duration)

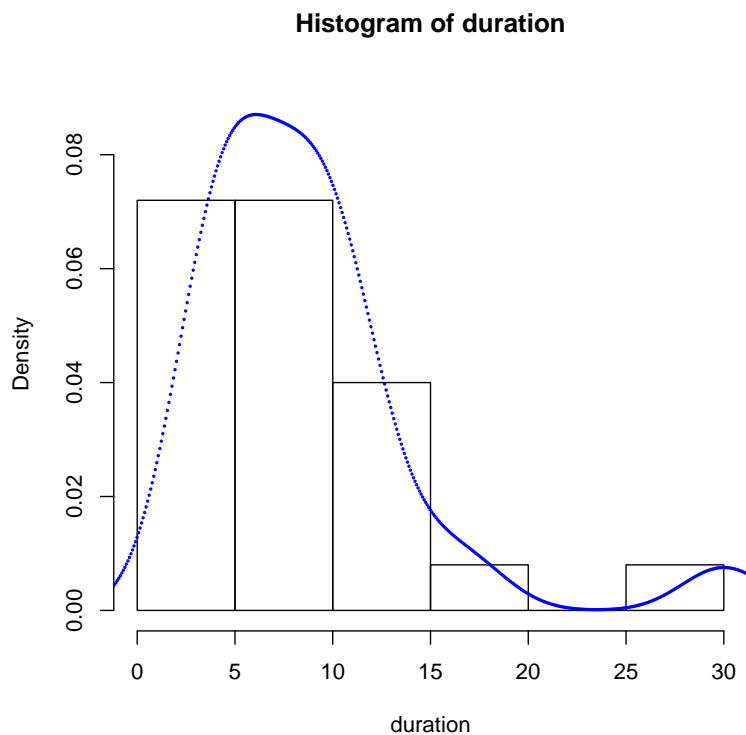
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.0    5.0    8.0    8.6   11.0   30.0

> range(duration)

[1] 3 30

> hist(duration, freq = FALSE, ylim = c(0, 0.09))
> points(density(duration), cex = 0.2, col = "blue")

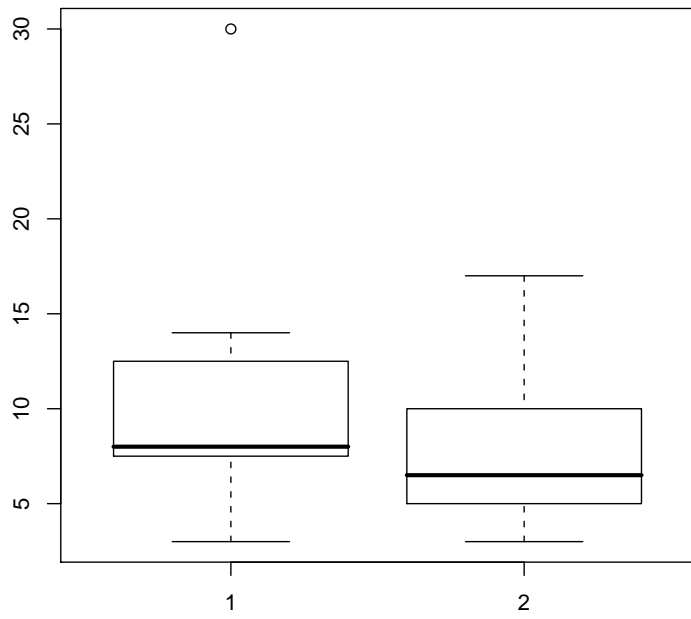
```



```

> antibiotics <- c(2, 2, 2, 2, 2, 1, 1, 2, 2, 2,
+ 2, 2, 1, 1, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2,
+ 2)
> noduration <- duration[antibiotics > 1.5]
> yesduration <- duration[antibiotics < 1.5]
> boxplot(duration ~ antibiotics)

```



```
> plot(duration ~ antibiotics)
```

