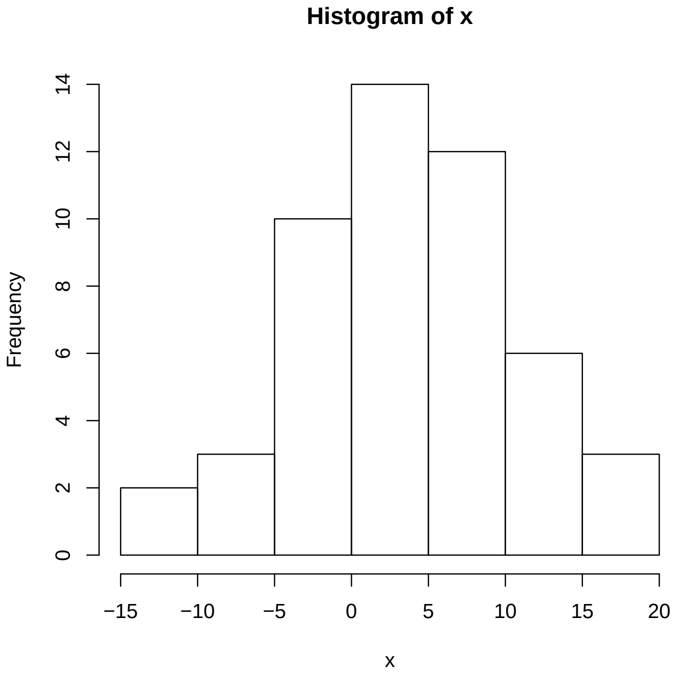
Data sets in R are either vectors or matrices. We begin by creating a vector and doing some elementary operations on it.

```
> w1 <- c(1, 2, 4)
> w1[3] <- 5
> w1

[1] 1 2 5

> w2 <- w1 + 4
> w2

[1] 5 6 9

> v <- c(w1, w2)
> v

[1] 1 2 5 5 6 9

> w1 * w2

[1]  5 12 45
```

Next we generate a random sample of size 50 from a normal population with mean 4 and standard deviation 7.

```
> x <- rnorm(50, 4, 7)
> x[1:4]

[1]  9.1785607  0.1166403 11.5567515  9.9926572

> x[c(1, 3)]

[1]  9.17856 11.55675

> sum(x)

[1] 198.434

> mean(x)

[1] 3.96868

> var(x)

[1] 51.07455

> min(x)

[1] -12.38789
```



```
> max(x)

[1] 19.85385

> quantile(x)

         0%         25%         50%         75%        100%
-12.3878933  -0.8600965   3.9775808   9.3459541  19.8538456

> quantile(x, c(0.45, 0.6))

     45%      60%
3.141016 6.040441
```

You can make lots of plots. For example, this histogram of the random sample of size 50 from the normal distribution was created by the command

```
> hist(x)
```

In the next bit of code we will generate observations from 5 independent binomial distributions and play around with some matrix notation.

```
> N <- c(10, 20, 30, 40, 50)
> p <- seq(0.1, 0.9, length = 5)
> y <- rbinom(5, N, p)
> M1 <- rbind(N, y)
> M1
```

```
  [,1] [,2] [,3] [,4] [,5]
N   10   20   30   40   50
y    1    5   21   28   41
```

```
> dim(M1)
```

```
[1] 2 5
```

```
> M1[2, ]
```

```
[1]  1  5 21 28 41
```

```
> M1[2, 5]
```

```
 y 
41 
```

```
> apply(M1, 1, mean)
```

```
   N    y 
30.0 19.2 
```

The function sample allows one to take random samples from a vector.

```
> sample(1:20, 5)
```

```
[1]  3 11 15  5  7
```

Note the command ?sample will give you more information about how the function sample works.

One nice thing about R is that it is easy to write functions to compute quantities of interest. The following simple example extracts the last value of a vector and its maximum.

```
> foo <- function(x) {
+     n <- length(x)
+     ans1 <- x[n]
+     ans2 <- max(x)
+     ans <- c(ans1, ans2)
+     return(ans)
+ }
> x <- c(1, 2, 3, 4, 5, 17, 0)
> foo(x)
```

```
[1]  0 17
```

As a final example we will write a function that allows you to take repeated random samples of size n from a population and calculate the mean of each sample.

```
> simmn <- function(y, n, W) {
+     ans <- rep(0, W)
+     for (i in 1:W) {
+         dum <- sample(y, n)
+         ans[i] <- mean(dum)
+     }
+     return(ans)
+ }
> y <- rgamma(500, 2)
> out <- simmn(y, 20, 10)
> out
```

```
 [1] 2.293480 2.229003 2.094842 2.323238 2.004828 1.723921 2.208257 1.578589
 [9] 1.950921 1.825869
```

```
> round(out, digits = 2)
```

```
 [1] 2.29 2.23 2.09 2.32 2.00 1.72 2.21 1.58 1.95 1.83
```