

## C03-3 Coercing data types

Coercing data elements into one of the basic R data types is straight forward. Just add “as.” to the data type and you have the name of the respective function.

```
x <- c(1.4, 2.5, 3.6, 4.0)
class(x)
```

```
## [1] "numeric"
```

```
as.character(x)
```

```
## [1] "1.4" "2.5" "3.6" "4"
```

```
as.integer(x)
```

```
## [1] 1 2 3 4
```

```
as.factor(x)
```

```
## [1] 1.4 2.5 3.6 4
## Levels: 1.4 2.5 3.6 4
```

Please note that the `as.integer` function returns the floor of the floating point values and does not apply any rounding rules.

So far, we have differentiated between data types and structures. Actually, a vector, list or data frame is nothing but another data type and the same logic of conversion functions can be used.

```
x <- c(1.4, 2.5, 3.6, 4.0)
class(x) # i.e. vector of type numeric
```

```
## [1] "numeric"
```

```
as.list(x)
```

```
## [[1]]
## [1] 1.4
##
## [[2]]
## [1] 2.5
##
## [[3]]
## [1] 3.6
##
## [[4]]
```

```
## [1] 4
```

```
as.data.frame(x)
```

```
##      x  
## 1 1.4  
## 2 2.5  
## 3 3.6  
## 4 4.0
```

```
as.matrix(x)
```

```
##      [,1]  
## [1,] 1.4  
## [2,] 2.5  
## [3,] 3.6  
## [4,] 4.0
```