

Chapter 14 Answers

Highlights

- Review these functions
 - is (testing)
 - as (coercing)

14.1 Write an R function named `Str2Num` which accepts a single argument which is a character vector whose elements are single digits. The function should convert this vector of characters to a numeric value. Test your function with the R command

```
> Str2Num(c("3","6","5"))
```

which should return

365

```
Str2Num = function (x) {  
  # return(sum(10^(rev(1:length(x))-1) * as(x,"numeric")))  
  return(  
    sum(  
      10^(  
        length(x):1-1  
      )  
      *  
      as(x,"numeric")  
    )  
  )  
  
  # in the test case:  
  # 10^2 * 3  
  # + 10^1 * 6  
  # + 10^0 * 5  
}  
Str2Num(c("3","6","5"))
```

```
## [1] 365
```

14.2 What is returned by the following R command ?

```
is.character(5 - 3i)
```

```
## [1] FALSE
```

14.3 What is returned by the following R command ?

```
x = 3 + 4i  
as.integer(x * Conj(x))
```

```
## [1] 25
```

14.4 Write an R function named `benford` (after Benford's Law) that returns the leading digit of its single numeric argument. The leading digit of 365 is 3; the leading digit of 0.0243 is 2.

```
benford = function(x) {  
  y <- abs(x) # eliminate negative signs  
  while ( y < 1 ) y <- y * 10 # eliminate leading zeroes  
  return(  
    y
```

```

        suppressWarnings(
          as.numeric(
            unlist(
              strsplit(
                as.character(y), ''
              )
            )
          )
        )
      )
    )
  }
  benford(365)

```

```
## [1] 3
```

```
benford(0.0243)
```

```
## [1] 2
```

```
benford(-17)
```

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

```
benford(-pi/1000)
```

```
## [1] 3
```

```

benford_ = function(x) {
  y <- abs(x)
  while ( y < 10 ) y <- y * 10
  while ( y > 100 ) y <- y / 10
  return ( y %/% 10 )
}

```

```
benford_(365)
```

```
## [1] 3
```

```
benford_(0.0243)
```

```
## [1] 2
```

```
benford_(-17)
```

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

```
benford_(-pi/1000)
```

```
## [1] 3
```

```

benford__ = function(x) {
  y <- unlist(strsplit(as.character(x), ''))
  z <- ifelse(y=="-", "0", ifelse(y==".", "0", y))
  zz <- as.numeric(z)
  return(zz[zz>0][1])
}

```

```
benford__(365)
```

```
## [1] 3
```

```
benford__(0.0243)
```

```
## [1] 2
```

```
benford__(-17)
```

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

```
benford__(-pi/1000)
```

```
## [1] 3
```

```
benford___ = function(x) {  
  x <- abs(x)           # deal with negative numbers  
  while(x < 1) x <- x * 10 # deal with smallish numbers  
  return(  
    as.numeric(  
      substr(  
        as.character(x) # convert number to character vector, length=1  
        ,1               # start = 1  
        ,1               # stop = 1  
      )  
    )  
  )  
}
```

```
benford___(365)
```

```
## [1] 3
```

```
benford___(0.0243)
```

```
## [1] 2
```

```
benford___(-17)
```

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

```
benford___(-pi/1000)
```

```
## [1] 3
```

14.5 Guess the effect of automatic coercion in the following R commands:

```
c(0,TRUE)
```

```
## [1] 0 1
```

```
c(2,FALSE)
```

```
## [1] 2 0
```

```
c(2L, 5)
```

```
## [1] 2 5
```

```
c(3L, 4.4)
```

```
## [1] 3.0 4.4
```

```
c(2.3, NA)
```

```
## [1] 2.3 NA
```

```
c(3, "R")

## [1] "3" "R"
c(FALSE, 9L, 4.4, "S")

## [1] "FALSE" "9"      "4.4"    "S"
class(c(0, TRUE))

## [1] "numeric"
class(c(2, FALSE))

## [1] "numeric"
class(c(2L, 5))

## [1] "numeric"
class(c(3L, 4.4))

## [1] "numeric"
class(c(2.3, NA))

## [1] "numeric"
class(c(3, "R"))

## [1] "character"
class(c(FALSE, 9L, 4.4, "S"))

## [1] "character"
```