Chapter 12 Answers

Highlights

- Data Types
  - NULL
  - logical
  - integer
  - numeric
  - complex
  - character
- Review these functions
  - any
  - all
  - ! (not, vectorized)
  - & (and, vectorized)
  - | (or, vectorized)
  - && (and, not vectorized)
  - || (or, not vectorized)
- Don't forget these functions
  - class
- Also introduced this chapter

12.1 Write a one-line function named `oddjob` with a single argument `x` that returns the elements in odd-numbered positions in the vector `x`.

```
# this is okay, but it doesn't use any of the cool new toys
oddjob = function(x) { return(x[seq(1,length(x),by=2)]) }
# this, however, uses a logical (true/false) element to indicate whether or not
# the element resting in that position should be returned
# see page 84, Section 12.2 in the text for a fuller explanation
oddjob_    = function(x) { return(x[  (1:length(x) %% 2 ) == TRUE ] ) }
# variations on the theme:
oddjob__   = function(x) { return(x[  (1:length(x) %% 2 ) != FALSE] ) }
oddjob___  = function(x) { return(x[!((1:length(x) %% 2 ) != TRUE )]) }
oddjob____ = function(x) { return(x[!((1:length(x) %% 2 ) == FALSE)]) }

x = 101:115
oddjob(x)
```

```
## [1] 101 103 105 107 109 111 113 115
```

```
oddjob_(x)
```

```
## [1] 101 103 105 107 109 111 113 115
```

```
oddjob__(x)
```

```
## [1] 101 103 105 107 109 111 113 115
```

```
oddjob___(x)
```

```
## [1] 101 103 105 107 109 111 113 115
```

```
oddjob____(x)
```

```
## [1] 101 103 105 107 109 111 113 115
```