1. Fundamental Concepts

In R, even a single number is a vector of length 1.

R

```r
# A scalar is just a vector of length 1
x <- 42
length(x)
# [1] 1
```

2. Creating Vectors

Common methods to initialize vectors.

R

```r
# Using c() to combine values
weights <- c(60.5, 72.3, 80.0)

# Creating a sequence from 1 to 10
counting <- 1:10

# Using seq() for specific steps
odds <- seq(from = 1, to = 10, by = 2)
# Result: 1, 3, 5, 7, 9

# Using rep() for patterns
repeated_pattern <- rep(c(1, 2), times = 3)
# Result: 1, 2, 1, 2, 1, 2
```

3. Subsetting Vectors

Accessing data within a vector using `[]`.

R

```r
v <- c(10, 20, 30, 40, 50)

# By Position: Get the first and third elements
v[c(1, 3)]
# [1] 10 30

# By Negative Index: Remove the second element
v[-2]
# [1] 10 30 40 50

# By Logic: Get values greater than 25
v[v > 25]
# [1] 30 40 50
```

4. Vectorized Operations & Recycling

Operations are applied to every element, and shorter vectors repeat to match longer ones.

R

```r
# Element-wise addition
a <- c(1, 2, 3)
b <- c(10, 10, 10)
a + b
# [1] 11 12 13

# Recycling Rule: c(1, 2) is recycled to c(1, 2, 1, 2)
short <- c(1, 2)
long <- c(10, 20, 30, 40)
short + long
# [1] 11 22 31 42
```

5. Vector Attributes (Names)

Adding metadata to your vectors.

R

```r
# Creating a named vector
temps <- c(Monday = 22, Tuesday = 25, Wednesday = 21)

# Accessing by name
temps["Tuesday"]
# Tuesday
#      25
```

6. Coercion (Type Conversion)

Converting between types using `as.*` functions.

R

```r
# Character to Numeric
char_vec <- c("1", "2", "3")
num_vec <- as.numeric(char_vec)

# Numeric to Logical (0 is FALSE, others are TRUE)
logic_vec <- as.logical(c(0, 1, 5))
# [1] FALSE  TRUE  TRUE
```