

# How can the setdiff function be used in R? Can you provide some examples?

Authored by  
**stats writer**

April 25, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can the setdiff function be used in R? Can you provide some examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=139123>

The setdiff function in R is a useful tool for finding the difference between two sets of data. It takes two input sets and returns a new set containing elements that are present in the first set but not in the second set. This function is commonly used in data analysis and manipulation to identify unique values or outliers.

To use the setdiff function, the two sets of data must be of the same data type, such as vectors or lists. It is also important to note that the order of the input sets is significant, as the returned set will contain elements from the first set in the same order.

For example, if we have two vectors, x and y, with the following values:

```
x = c(1, 2, 3, 4, 5)
```

```
y = c(3, 5, 7, 9)
```

Using the setdiff function, we can find the elements in x that are not present in y:

```
setdiff(x, y)
```

```
Output: 1, 2, 4
```

Similarly, we can also use the setdiff function to identify unique values in a data frame. For instance, if we have a data frame with two columns, "ID" and "Name", and we want to find the unique IDs that are not present in another data frame, we can use the setdiff function as follows:

```
uniqueIDs = setdiff(df1$ID, df2$ID)
```

In summary, the setdiff function in R is a useful tool for comparing and identifying differences between two sets of data. It is versatile and can be used in various scenarios, making it an essential function in data analysis and manipulation.

## Use the setdiff Function in R (With Examples)

**The setdiff() function in R can be used to find differences between two sets. This function uses the following syntax:**

```
setdiff(x, y)
```

**where:**

**x, y: Vectors or data frames containing a sequence of items**

**This tutorial provides several examples of how to use this function in practice.**

#### **Example 1: Setdiff with Numeric Vectors**

**The following code shows how to use setdiff() to identify all of the values in vector *a* that do not occur in vector *b*:**

```
#define vectors
```

```
a <- c(1, 3, 4, 5, 9, 10)
```

```
b <- c(1, 2, 3, 4, 5, 6)
```

```
#find all values in a that do not occur in bsetdiff(a, b)
```

```
9 10
```

**There are two values that occur in vector *a* that do not occur in vector *b*: 9 and 10.**

**If we reverse the order of the vectors in the setdiff() function, we can instead identify all of the values in vector *b* that do not occur in vector *a*:**

**#find all values in *b* that do not occur in a**  
**setdiff(b, a)**

**2 6**

**There are two values that occur in vector *b* that do not occur in vector *a*: 2 and 6.**

**Example 2: Setdiff with Character Vectors**

**The following code shows how to use setdiff() to identify all of the values in vector *char1* that do not occur in vector *char2*:**

```
#define character vectors  
char1 <- c('A', 'B', 'C', 'D', 'E')  
char2 <- c('A', 'B', 'E', 'F', 'G')
```

```
#find all values in char1 that do not occur in  
char2  
setdiff(char1, char2)
```

```
"C" "D"
```

**Example 3: Setdiff with Data Frames**

**The following code shows how to use setdiff() to identify all of the values in one data frame column that do not appear in the same column of a second data**

**frame:**

**#define data frames**

```
df1 <- data.frame(team=c('A', 'B', 'C', 'D'),  
conference=c('West', 'West', 'East', 'East'),  
points=c(88, 97, 94, 104))
```

```
df2 <- data.frame(team=c('A', 'B', 'C', 'D'),  
conference=c('West', 'West', 'East', 'East'),  
points=c(88, 97, 98, 99))
```

**#find differences between the points columns in the two data frames**

```
setdiff(df1$points, df2$points)
```

**94 104**

**We can see that the values 94 and 104 occur in the points column of the first data frame, but not in the points column of the second data frame.**

## **How to Perform Partial String Matching in R**