

How can I use dplyr to recode values?

Authored by
stats writer

April 17, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use dplyr to recode values?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=136424>

Dplyr is a powerful data manipulation package in R that allows users to easily recode values in their datasets. By using the "mutate" function, users can create new columns and manipulate existing ones, including changing values based on certain criteria. This allows for efficient and precise recoding of data without the need for manual coding. Additionally, dplyr's intuitive syntax makes it easy to understand and implement, making it a valuable tool for data analysis and cleaning. With dplyr, users can quickly and effectively recode values in their datasets, saving time and ensuring accurate data analysis.

Recode Values Using dplyr

Occasionally you may be interested in recoding certain values in a dataframe in R. Fortunately this can easily be done using the function from the dplyr package.

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

Example 1: Recode a Single Column in a Dataframe

The following code shows how to recode a single column in a dataframe:

```
library(dplyr)#create dataframe
df <- data.frame(player = c('A', 'B', 'C', 'D'),
points = c(24, 29, 13, 15),
result = c('Win', 'Loss', 'Win', 'Loss'))
```

```
#view dataframe
```

```
df
```

```
#change 'Win' and 'Loss' to '1' and '0'  
df %>% mutate(result=recode(result, 'Win'='1',  
'Loss'='0'))
```

```
player points result
```

```
1 A 24 1
```

```
2 B 29 0
```

```
3 C 13 1
```

```
4 D 15 0
```

Example 2: Recode a Single Column in a Dataframe and Provide NA Values

The following code shows how to recode a single column in a dataframe and give a value of NA to any value that is not explicitly given a new value:

```
library(dplyr)  
#create dataframe  
df <- data.frame(player = c('A', 'B', 'C', 'D'),  
points = c(24, 29, 13, 15),  
result = c('Win', 'Loss', 'Win', 'Loss'))
```

```
#view dataframe
```

```
df
```

```
#change 'Win' to '1' and give all other values a value of
```

NA

```
df %>% mutate(result=recode(result, 'Win'='1',  
.default=NA_character_))
```

player points result

1 A 24 1

2 B 29 <NA>

3 C 13 1

4 D 15 <NA>

Example 3: Recode Multiple Columns in a Dataframe

The following code shows how to recode multiple columns at once in a dataframe:

```
library(dplyr)
```

```
#create dataframe
```

```
df <- data.frame(player = c('A', 'B', 'C', 'D'),
```

```
points = c(24, 29, 13, 15),
```

```
result = c('Win', 'Loss', 'Win', 'Loss'))
```

```
#recode 'player' and 'result' columns
```

```
df %>% mutate(player=recode(player, 'A'='Z'),
```

```
result=recode(result, 'Win'='1', 'Loss'='0'))
```

player points result

1 Z 24 1

2 B 29 0

3 C 13 1

4 D 15 0

You can find the complete documentation for the `recode()` function .

ARABPSYCHOLOGY.COM