How can I perform fuzzy matching in R, and can you provide an example?

Authored by

stats writer

June 29, 2024

Fuzzy matching is a technique used in data analysis to identify and match similar strings or words, even if they are not an exact match. In R, fuzzy matching can be performed using the package "stringdist" which offers various functions for measuring the distance between strings and finding the best match. An example of fuzzy matching in R could involve merging two datasets with slightly different spellings of names, where the "stringdist" function can be used to find and match the most similar names between the two datasets. This allows for more accurate and comprehensive data analysis.

Perform Fuzzy Matching in R (With Example)

Often you may want to join together two datasets in R based on imperfectly matching strings. This is sometimes called fuzzy matching.

The easiest way to perform fuzzy matching in R is to use the stringdist_join() function from the fuzzyjoin package.

The following example shows how to use this function in practice.

Example: Fuzzy Matching in R

Suppose we have the following two data frames in R that contain information about various basketball teams:

```
#create data frames
df1 <- data.frame(team=c('Mavericks', 'Nets', 'Warriors',
```

```
'Heat', 'Lakers'),
points=c(99, 90, 104, 117, 100))
df2 <- data.frame(team=c('Mavricks', 'Warrors', 'Heat',
'Netts', 'Kings', 'Lakes'),
assists=c(22, 29, 17, 40, 32, 30))

#view data frames
print(df1)

team points
1 Mavericks 99
2 Nets 90
3 Warriors 104
4 Heat 117
5 Lakers 100

print(df2)

team assists
1 Mavricks 22
2 Warrors 29
3 Heat 17
4 Netts 40
5 Kings 32
6 Lakes 30
```

Now suppose that we would like to perform a left join in which we keep all of the rows from the first data frame and simply merge them based on the team name that most closely matches in the second data frame.

We can use the following code to do so:

```
library(fuzzyjoin)
library(dplyr)

#perform fuzzy matching left join
stringdist_join(df1, df2,
by='team', #match based on team
mode='left', #use left join
method = "jw", #use jw distance metric
max_dist=99,
distance_col='dist') %>%
group_by(team.x) %>%
slice_min(order_by=dist, n=1)

# A tibble: 5 x 5
# Groups: team.x
team.x points team.y assists dist

1 Heat 117 Heat 17 0
```

2 Lakers 100 Lakes 30 0.0556

3 Mavericks 99 Mavricks 22 0.0370

4 Nets 90 Netts 40 0.0667

5 Warriors 104 Warrors 29 0.0417

The result is one data frame that contains each of the five original team names from the first data frame along with the team that most closely matches from the second data frame.

Note #1: We chose to use the jw distance metric for matching. This is short for the , which is a metric that measures the difference between two strings.

Note #2: We used the slice_min() function from the dplyr package to only show the team name from the second data frame that most closely matched the team name from the first data frame.

Additional Resources

The following tutorials explain how to perform other common tasks in R: