Numbering rows within groups in a data frame

Ask Question

Working with a data frame similar to this:

```
set.seed(100)
df <- data.frame(cat = c(rep("aaa", 5), rep("bbb", 5), rep("ccc", 5)), val = runif(15))
df <- df[order(df$cat, df$val), ]
df

   cat        val
1  aaa 0.05638315
2  aaa 0.25767250
3  aaa 0.30776611
4  aaa 0.46854928
5  aaa 0.55232243
6  bbb 0.17026205
7  bbb 0.37032054
8  bbb 0.48377074
9  bbb 0.54655860
10 bbb 0.81240262
11 ccc 0.28035384
12 ccc 0.39848790
13 ccc 0.62499648
14 ccc 0.76255108
15 ccc 0.88216552
```

I am trying to add a column with numbering within each group. Doing it this way obviously isn't using the powers of R:

```
 df$num <- 1
 for (i in 2:(length(df[,1]))) {
   if (df[i,"cat"]==df[(i-1),"cat"]) {
     df[i,"num"]<-df[i-1,"num"]+1
     }
 }
 df

   cat        val num
1  aaa 0.05638315   1
2  aaa 0.25767250   2
3  aaa 0.30776611   3
4  aaa 0.46854928   4
5  aaa 0.55232243   5
6  bbb 0.17026205   1
7  bbb 0.37032054   2
8  bbb 0.48377074   3
9  bbb 0.54655860   4
10 bbb 0.81240262   5
```

What would be a good way to do this?

r dataframe r-faq

edited Oct 6 '17 at 19:53
Jaap
53k ● 20 ● 115 ● 123

asked Oct 16 '12 at 23:38
eli-k
4,117 ● 7 ● 28 ● 39

5 Answers

Use `ave`, `ddply`, `dplyr` or `data.table`:

```
df$num <- ave(df$val, df$cat, FUN = se
```

or:

```
library(plyr)
ddply(df, .(cat), mutate, id = seq_alc
```

or:

```
library(dplyr)
df %>% group_by(cat) %>% mutate(id = r
```

or (the most memory efficient, as it assigns by reference within `DT`):

```
library(data.table)
DT <- data.table(df)

DT[, id := seq_len(.N), by = cat]
DT[, id := rowid(cat)]
```

edited Mar 14 '17 at 22:06
Frank
52.9k ● 6 ● 53 ● 120

answered Oct 16 '12 at 23:41
mnel
88.8k ● 14 ● 213 ● 221

1 It might be worth mentioning that `ave` gives a float instead of an int here.

1 Interestingly this `data.table` solution seems to be quicker than using `frank` : `library(microbenchmark); microbenchmark(a = DT[, .(val ,num = frank(val)), by = list(cat)] ,b =DT[, .(val , id = seq_len(.N)), by = list(cat)] , times = 1000L)` – hannes101 Jul 28 '17 at 12:23

Thanks! The `dplyr` solution is good. But if, like me, you kept getting weird errors when trying this approach, make sure that you are not getting conflicts between `plyr` and `dplyr` as explained in this post It can be avoided by explicitly calling `dplyr::mutate(...)` – EcologyTom Apr 10 at 14:16

1 another `data.table` method is `setDT(df)[, id:=rleid(val), by=.(cat)]` – chinsoon12 May 23 at 0:14

How to modify `library(plyr)` and `library(dplyr)` answers to make the ranking val column in descending order? – Przemyslaw Remin Jul 24 at 9:31

For making this r-faq question more complete, a base R alternative with `sequence` and `rle` :

```
df$num <- sequence(rle(df$cat)$lengths
```

which gives the intended result:

```
> df
   cat        val num
4  aaa 0.05638315   1
2  aaa 0.25767250   2
1  aaa 0.30776611   3
5  aaa 0.46854928   4
3  aaa 0.55232243   5
10 bbb 0.17026205   1
8  bbb 0.37032054   2
6  bbb 0.48377074   3
9  bbb 0.54655860   4
7  bbb 0.81240262   5
13 ccc 0.28035384   1
14 ccc 0.39848790   2
11 ccc 0.62499648   3
15 ccc 0.76255108   4
12 ccc 0.88216552   5
```

answered Oct 6 '17 at 20:01

Jaap
53k ● 20 ● 115 ● 123

Here is an option using a `for` loop by groups rather by rows (like OP did)

```
for (i in unique(df$cat)) df$num[df$ca
```

edited Feb 26 '17 at 8:53

David Arenburg
76.9k ● 10 ● 90 ● 155

answered Oct 16 '12 at 23:51

alittleboy
4,159 ● 11 ● 39 ● 89

I would like to add a `data.table` variant using the `rank()` function which provides the additional possibility to change the ordering and thus makes it a bit more flexible than the `seq_len()` solution and is pretty similar to row_number functions in RDBMS.

```
# Variant with ascending ordering
library(data.table)
dt <- data.table(df)
dt[, .( val
   , num = rank(val))
    , by = list(cat)][order(cat, num),

    cat        val num
 1: aaa 0.05638315   1
 2: aaa 0.25767250   2
 3: aaa 0.30776611   3
 4: aaa 0.46854928   4
 5: aaa 0.55232243   5
 6: bbb 0.17026205   1
 7: bbb 0.37032054   2
 8: bbb 0.48377074   3
 9: bbb 0.54655860   4
10: bbb 0.81240262   5
11: ccc 0.28035384   1
12: ccc 0.39848790   2
13: ccc 0.62499648   3
14: ccc 0.76255108   4

# Variant with descending ordering
```

answered Jun 18 at 9:28

hannes101
596 ● 5 ● 15

Here is a small improvement trick that allows sort 'val' inside the groups:

```
# 1. Data set
set.seed(100)
df <- data.frame(
  cat = c(rep("aaa", 5), rep("ccc", 5)
  val = runif(15))

# 2. 'dplyr' approach
df %>%
  arrange(cat, val) %>%
  group_by(cat) %>%
  mutate(id = row_number())
```

edited Sep 22 at 8:06

answered Sep 22 at 7:40

andrii
763 ● 8 ● 10

protected by zx8754 May 23 at 6:52

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?