Data Frames II

Santiago Lozano

21 de marzo de 2020

Usar order() y !duplicated() para eliminar pseudoreplicación

En un ejemplo más complicado quisieramos extraer una observación de cada tipo de vegetación, y que si es así que extraiga el de worm density mayor. Hay 2 pasos par hacer, el primero es ordenar las filas y el segundo es seleccionar los no duplicados, así

```
setwd("C:/Users/santiago/Documents/Progrmación en R/2020-I/PR08-Data Frames")
worms <- read.table("worms.txt",header = T,dec = ".")
attach(worms)
```

```
new <- worms[rev(order(Worm.density)),]
new[!duplicated(new$Vegetation),]
```

```
##        Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 9     The.Orchard  1.9     0    Orchard     5.7 FALSE            9
## 16   Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
## 11    Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
## 10  Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
## 2  Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
```

Ordenamiento complejo

Existen ocasiones em la que queremos order varias variables pero ordenarlas en sentidos opuestos

```
worms[order(Vegetation,-Worm.density),]
```

```
##           Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
## 18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
## 8            Ashurst  2.1     0     Arable     4.8 FALSE            4
## 10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
## 1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
## 7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
## 3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
## 6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
## 13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
## 12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
## 19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
## 14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
## 16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
## 15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
## 4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
## 9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
## 11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
## 5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
## 17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
## 20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

Aquí Vegetation se orden alfabético, pero worm density se ordenó de mayor a menor

Para ordenar de manera contraria al alfabeto podemos usar la función rank()

```
worms[order(-rank(Vegetation),-Worm.density),]
```

```
##           Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
## 5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
## 17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
## 20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
## 9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
## 16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
## 15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
## 4         Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
## 10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
## 1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
## 7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
## 3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
## 6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
## 13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
## 12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
## 19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
## 14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
## 2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
## 18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
## 8            Ashurst  2.1     0     Arable     4.8 FALSE            4
```

Una forma menos probable de ordenar será seleccionando columna sobre operadores lógico. Suponga que por alguna razón usted quiere seleccionar las columnas que contienen el caracter “S”, así

```
names(worms)
```

```
## [1] "Field.Name"   "Area"         "Slope"        "Vegetation"
## [5] "Soil.pH"      "Damp"         "Worm.density"
```

```
grep("S",names(worms))
```

```
## [1] 3 5
```

de esta manera

```
worms[,grep("S",names(worms))]
```

```
##    Slope Soil.pH
## 1     11     4.1
## 2      2     5.2
## 3      3     4.3
## 4      5     4.9
## 5      0     4.2
## 6      2     3.9
## 7      3     4.2
## 8      0     4.8
## 9      0     5.7
## 10     4     5.0
## 11    10     5.2
## 12     1     4.1
## 13     2     4.0
```

```
## 14     6     3.8
## 15     0     5.0
## 16     0     4.9
## 17     8     4.7
## 18     2     4.5
## 19     1     3.5
## 20    10     5.1
```

Un Dataframe con nombres de filas en vez de número de fila

Podemos suprimir la creación de número de filas, ubicando su propio nombre de cada fila alteranndo la sintaxis de read.table

```
worms2 <- read.table("worms.txt",header = T,dec = ".",row.names = 1)
worms2
```

```
##                   Area Slope Vegetation Soil.pH  Damp Worm.density
## Nashs.Field        3.6    11  Grassland     4.1 FALSE            4
## Silwood.Bottom     5.1     2     Arable     5.2 FALSE            7
## Nursery.Field      2.8     3  Grassland     4.3 FALSE            2
## Rush.Meadow        2.4     5     Meadow     4.9  TRUE            5
## Gunness.Thicket    3.8     0      Scrub     4.2 FALSE            6
## Oak.Mead           3.1     2  Grassland     3.9 FALSE            2
## Church.Field       3.5     3  Grassland     4.2 FALSE            3
## Ashurst            2.1     0     Arable     4.8 FALSE            4
## The.Orchard        1.9     0    Orchard     5.7 FALSE            9
## Rookery.Slope      1.5     4  Grassland     5.0  TRUE            7
## Garden.Wood        2.9    10      Scrub     5.2 FALSE            8
## North.Gravel       3.3     1  Grassland     4.1 FALSE            1
## South.Gravel       3.7     2  Grassland     4.0 FALSE            2
## Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
## Pond.Field         4.1     0     Meadow     5.0  TRUE            6
## Water.Meadow       3.9     0     Meadow     4.9  TRUE            8
## Cheapside          2.2     8      Scrub     4.7  TRUE            4
## Pound.Hill         4.4     2     Arable     4.5 FALSE            5
## Gravel.Pit         2.9     1  Grassland     3.5 FALSE            1
## Farm.Wood          0.8    10      Scrub     5.1  TRUE            3
```

Crear un dataframe a partir de otro tipo de objeto

Sabemos que podemos usar la función table() para contar los elementos de un vector

```
y <- rpois(1500,1.5)
table(y)
```

```
## y
##   0   1   2   3   4   5   6   7
## 340 465 405 192  70  22   4   2
```

Y Podemos este simple conteo convertirlo en un dataframe

```
short<-as.data.frame(table(y))
short
```

```
##   y Freq
```

```
## 1 0  340
## 2 1  465
## 3 2  405
## 4 3  192
## 5 4   70
## 6 5   22
## 7 6    4
## 8 7    2
```

```
index<-rep(1:8,short$Freq)
index
```

```
##    [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##   [35] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##   [69] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [103] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [137] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [171] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [205] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [239] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [273] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [307] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [341] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [375] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [409] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [443] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [477] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [511] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [545] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [579] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [613] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [647] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [681] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [715] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [749] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [783] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3
##  [817] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [851] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [885] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [919] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [953] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [987] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [1021] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [1055] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [1089] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [1123] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [1157] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [1191] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4
## [1225] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1259] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1293] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1327] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1361] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1395] 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [1429] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
```

```
## [1463] 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7
## [1497] 7 7 8 8
```