Aggregieren mit data.table in R

Benjamin Schlegel

28. November 2016

In diesem Artikel wird erklärt, wie man mit Hilfe der Bibliothek data.table schnell und einfach aggregieren kann. Als Beispiel werden Daten der Mondial Datenbank verwendet. Die Daten wurden 1998 generiert.

In einem ersten Schritt werden die Daten aus der MySQL Datenbank dieses Servers heruntergeladen. Eine Erklärung des folgendes Codes würde den Rahmen dieses Artikel sprengen und nichts zum Verstehen von data.table beitragen.

```
library(RMySQL)
db = dbConnect(MySQL(), user='solidari_mondo', password='ZL8H8KZD',
  dbname='solidari_mondial', host='solidari.mysql.db.hostpoint.ch')
sql = "SELECT Name as Country, Continent, GDP/Population*1000000 as GDPPC, Population
       FROM country JOIN economy JOIN encompasses
       ON economy.Country = country.Code AND encompasses.Country = country.Code";
rs = dbSendQuery(db, sql)
data = fetch(rs, n=-1)
data = na.omit(data)
```

Nun sind die Daten in R eingelesen. Als erstes wird das data.frame in ein data.table-Objekt umgewandelt.

```
library(data.table)
data.t = as.data.table(data)
```

Nun kann mit dem Objekt gearbeitet werden. Dazu werden eckige Klammern verwendet. Zwischen den Klammern hat es drei Argumente. Das erste kann zum Filtern verwendet werden, das zweite zum Auswählen und das dritte Argument zum Gruppieren. Wir wollen den durchschnittlichen BIP pro Kopf und die Bevölkerung jedes Kontinents berechnen. Da wir zwei Werte im zweiten Argument wollen, verwenden wir .(). Bei dritten Argument geben wir by=Continent. Auch hier könnten mit .() mehrere Variablen angegeben werden. Anschliessend wandeln wir es wieder in ein data.frame zurück.

```
data.agg = data.t[,.(avg.gdppc=mean(GDPPC),population=sum(Population)),by=Continent]
data.agg = as.data.frame(data.agg)
data.agg
```

```
          Continent avg.gdppc population
1            Europe 11788.929  791089680
2              Asia 14667.149 3712260290
3           America  9759.018  784226681
4 Australia/Oceania  7004.231  236373230
5            Africa  1825.906  730938619
```

data.table kann auch gut mit riesigen Datenmengen umgehen. So soll ein Datensatz der Grösse 500 GB keinerlei Probleme darstellen.