Daten laden und speichern

Dirk Seidensticker/Clemens Schmid

7. Juli 2017

Daten aus dem Dateisystem laden

Daten aus dem Dateisystem laden

```
df <- read.csv("../data/AtlantPottery.csv")
```

bei csv evtl. zu beachten:

- **sep =** identifiziert das Trennzeichen
  - `";"` Semikolon
  - `"\t"` Tabulator
- **dec = “,”** identifiziert das Komma als Dezimalzeichen
- **row.names = 1** definiert erste Spalte als Zeilennamen (index)

Daten aus dem Dateisystem laden

```
head(df)
```

```
##   site feature object class sherd qty   wt size wall muendungsD muen
## 1    C surface     54     K     G   1 1448  500    7        17.5
## 2    C surface     36     K     R   1   56  120    5        17.0
## 3    C surface     32     K     R   1   92  120    9        20.0
## 4    C surface     26     K     R   1   45   70   11        22.5
## 5    E surface     86     K     G   1 1298  500    8        18.5
## 6    E surface     63     K     G   1  667  200   NA         6.0
##   minD minD_H maxD maxD_H bodenD temperSize vesselShape
## 1 16.0   16.5 21.0   11.0    7.5          M           3
## 2 14.5     NA 16.0     NA     NA          M           7
## 3 18.5     NA 24.5     NA     NA         VF           7
## 4   NA     NA 27.0     NA     NA         VF           8
## 5 22.5    9.5 22.5    9.5    0.0         VC           8
## 6  5.5   11.5 10.5    5.5    6.0          C           1
```

Andere Datenquellen

Datenbanken und XML

- Datenbanken
  - **SQLite**: library(RSQLite)
  - **PostGreSQL**: library(RPostgreSQL)
  - MySQL, Oracle, ODBC (library(RODBC)), MongoDB, ...
- hierarchisch strukturierter Daten
  - **XML**
  - JSON

SQLite

```
# nicht lauffähig
library(RSQLite)
drv <- dbDriver("SQLite")
con <- dbConnect(drv, "../data/DB.sqlite")

df = dbGetQuery(con, "SELECT * FROM ...")
head(df)
```

PostgreSQL

```r
# nicht lauffähig
library(RPostgreSQL)
drv <- dbDriver("PostgreSQL")
con <- dbConnect(drv, dbname = "DATENBANKNAME",
                 host = "escience-center.uni-tuebingen.de",
                 port = 12345,
                 user = "user", password = "pw")

df = dbGetQuery(con, "SELECT * FROM ...")
head(df)
```

XML

```
library(XML)
library(RCurl)
```

```
## Loading required package: bitops
```

```
url <- "https://www.w3schools.com/xml/simple.xml"
data <- getURL(url)
doc <- xmlParse(data)
class(doc)
```

```
## [1] "XMLInternalDocument" "XMLAbstractDocument"
```

XML

```
doc <- xmlRoot(doc) #gives content of root
doc[1]
```

```
## $food
## <food>
##   <name>Belgian Waffles</name>
##   <price>$5.95</price>
##   <description>Two of our famous Belgian Waffles with plenty of real
##   <calories>650</calories>
## </food>
##
## attr(,"class")
## [1] "XMLInternalNodeList" "XMLNodeList"
```

```
df <- xmlToDataFrame(nodes=getNodeSet(doc,"//food"))
df[,c(1,2,4)]
```

```
##                          name price calories
## 1             Belgian Waffles $5.95      650
## 2  Strawberry Belgian Waffles $7.95      900
## 3 Berry-Berry Belgian Waffles $8.95      900
## 4                French Toast $4.50      600
## 5         Homestyle Breakfast $6.95      950
```