Dates and DBI Cheatsheet

Brad McNeney

2018-02-14

Dates

- Coerce a character vector `v` with dates specified by a characer string `format` to a `Date` object with:

  `as.Date(v,format)`

- The default formats for the date are `%Y-%m-%d` or `%Y/%m/%d`, where :
  - `%Y` is year with century (e.g., 2018)
  - `%m` is month,
  - `%d` is day,
  - separators are `-` or `/`.
- Can also specify `%y` for year without century (00-68 mean 2000-2068, 69-99 mean 1969-1999).
- Example non-standard date format is `"%m/%d/%y"` for month, day, years; e.g., `08/30/08` is August 30, 2008.

DBI and SQLite

- Initialize an SQLite database with

```
library(DBI)
mydb <- dbConnect(RSQLite::SQLite(), "my-db.sqlite")
```

- Insert a table with

  `dbWriteTable(mydb,name='tablename', value = mydf)`

  where `tablename` is the name for the table in the database and `mydf` is a data frame containing the table.

- Query with

  `dbGetQuery(wdb,query)`

  where `query` is a character string containing SQL statements.

- SQL query examples:
  - `SELECT var1, var2, var3 FROM table1`
  - `SELECT table1.var1, table2.var1 FROM table1 INNER JOIN table2 ON table1.ID=table2.ID`
  - `SELECT table1.var1, table2.var1 FROM table1 LEFT JOIN table2 ON table1.ID=table2.ID`

- Batched queries with

```
rs <- dbSendQuery(mydb,query)
while (!dbHasCompleted(rs)) {
  df <- dbFetch(rs, n = 2)
}
dbClearResult(rs)
```

- Parametrized queries with

```
rs <- dbSendQuery(mydb,"SELECT * FROM table WHERE var1 >= :x")
dbBind(rs,param = list(x=40))
dbFetch(rs)
dbClearResult(rs)
```

- Close the connection to the database and remove the database file with

```
dbDisconnect(mydb)
unlink("my-db.sqlite")
```