Big Data Statistics for R and Python

Lecture 4: Cleaning and Transformation of Big Data

Prof. Dr. Ulrich Matter
(University of St. Gallen)

11/03/2019

1 Cleaning and Transformation of large data sets

Practically preceeding the filtering of the data and the selection/aggregation of raw data in order to perform statistical analyses, the steps related to cleaning and transformation of the data typically have to be run on large parts of the totally available data. Typically the bottleneck is a lack of RAM. In the following, we explore two strategies that broadly build on the idea of *virtual memory* (using parts of the hard disk as RAM).

1.1 'Out-of-memory' strategies

Virtual memory is in simple words an approach to combining the RAM and Mass Storage components in order to cope with a lack of RAM. Modern operating systems come with a virtual memory manager that would automatically handle the swapping between RAM and the hard-disk, when runnign processes use up too much RAM. However, a virtual memory manager is not specifically developed to perform this task in the context of data analysis. Several strategies have thus been developed to build on the basic idea of virtual memory in the context of data analysis tasks.

- *Chunked data files on disk*: The data analytics software 'partitions' the large dataset, maps, and stores the chunks of raw data on disk. What is actually 'read' into RAM when importing the data file with this approach is the mapping to the partitions of the actual dataset (the data structure) and some metadata describing the dataset. In R, this approach is implemented in the `ff` package and several packages building on `ff`. In this approach, the usage of disk space and the linking between RAM and files on disk is very explicit (and well visible to the user).
- *Memory mapped files and shared memory*: The data analytics software uses segments of virtual memory for the dataset to be analyzed and allows different programs/processes to access it in the same memory segment. Thus, virtual memory is explicitly allocated for one or several specific data analytics tasks. In R, this approach is prominently implemented in the `bigmemory` package and several packages building on `bigmemory`.

1.1.1 Chunking data with the `ff`-package

Before looking at the more detailed and applied code examples in Walkowiak (2016), we investigate how the `ff` package (and the concept of chunked files) basically works. In order to do so, we first install and load the `ff` and `ffbase` packages, as well as the `pryr` package. We use the already known `flights.csv`-dataset as an example. When importing data via the `ff` package, we first have to set up a directory where `ff` can store the partitioned dataset (recall that this is explicitly/visibly done on disk). As in the code examples of the book, we call this new directory `ffdf` (after `ff`-data.frame).

```
# SET UP --------------

# install.packages(c("ff", "ffbase"))
# load packages
```

```
library(ff)
library(ffbase)
library(pryr)

# create directory for ff chunks, and assign directory to ff
system("mkdir ffdf")
options(fftempdir = "ffdf")
```

Now we can read in the data with `read.table.ffdf`. In order to better understand the underlying concept, we record the change in memory in the R environment with `mem_change()`.

```
mem_change(
flights <-
     read.table.ffdf(file="../data/flights.csv",
                     sep=",",
                     VERBOSE=TRUE,
                     header=TRUE,
                     next.rows=100000,
                     colClasses=NA)
)
```

```
## read.table.ffdf 1..100000 (100000)  csv-read=0.634sec ffdf-write=0.12sec
## read.table.ffdf 100001..200000 (100000)  csv-read=0.632sec ffdf-write=0.085sec
## read.table.ffdf 200001..300000 (100000)  csv-read=0.802sec ffdf-write=0.063sec
## read.table.ffdf 300001..336776 (36776)  csv-read=0.36sec ffdf-write=0.05sec
##  csv-read=2.428sec  ffdf-write=0.318sec  TOTAL=2.746sec
```

```
## 783 kB
```

Note that there are two substantial differences to what we have previously seen when using `fread()` to import a csv file. It takes much longer to import a csv into the ffdf structure. However, the RAM allocated to it is much smaller. This is exactly what we would expect, keeping in mind what `read.table.ffdf()` does in comparison to what `fread()` does.

Now we can actually have a look at the data chunks created by `ff`, as well as how the structure of the dataset is represented in the `flights` object.

```
# show the files in the directory keeping the chunks
list.files("ffdf")
```

```
##  [1] "ffdf1a5812daf8d4.ff" "ffdf1a5815b18eb4.ff" "ffdf1a5818e94e98.ff"
##  [4] "ffdf1a581908fd55.ff" "ffdf1a581d2fe44a.ff" "ffdf1a58200c78fc.ff"
##  [7] "ffdf1a5820373575.ff" "ffdf1a582a1a1f00.ff" "ffdf1a583334c53e.ff"
## [10] "ffdf1a5833cea022.ff" "ffdf1a583c0fd68c.ff" "ffdf1a58427332c0.ff"
## [13] "ffdf1a584d8183b5.ff" "ffdf1a585a20a6f6.ff" "ffdf1a585c84873.ff"
## [16] "ffdf1a5866062ff7.ff" "ffdf1a5871ee07d3.ff" "ffdf1a587d26e9ee.ff"
## [19] "ffdf1a58896a4d9.ff"
```

```
# investigate the structure of the object created in the R environment
str(flights)
```

```
## List of 3
##  $ virtual: 'data.frame':    19 obs. of  7 variables:
##  .. $ VirtualVmode     : chr  "integer" "integer" "integer" "integer" ...
##  .. $ AsIs             : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  .. $ VirtualIsMatrix  : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  .. $ PhysicalIsMatrix : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  .. $ PhysicalElementNo: int  1 2 3 4 5 6 7 8 9 10 ...
```

```
##  .. $ PhysicalFirstCol : int  1 1 1 1 1 1 1 1 1 1 ...
##  .. $ PhysicalLastCol  : int  1 1 1 1 1 1 1 1 1 1 ...
##  .. - attr(*, "Dim")= int  336776 19
##  .. - attr(*, "Dimorder")= int  1 2
##  $ physical: List of 19
##  .. $ year          : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ month         : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ day           : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ dep_time      : list()
```

```
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ sched_dep_time: list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ dep_delay     : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ arr_time      : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
```

```
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ sched_arr_time: list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ arr_delay     : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ carrier       : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
```

```
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:16] "9E" "AA" "AS" "B6" ...
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ flight        : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ tailnum       : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:4044] "" "N0EGMQ" "N10156" "N102UW" ...
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ origin        : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:3] "EWR" "JFK" "LGA"
```

```
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ dest          : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:105] "ABQ" "ACK" "ALB" "ATL" ...
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ air_time      : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ distance      : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ hour          : list()
```

```
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ minute        : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ time_hour     : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 336776
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 336776
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:6936] "2013-01-01T10:00:00Z" "2013-01-01T11:00:00Z" "2013-01-0
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
##  ..  .. ..- attr(*, "class")= chr "virtual"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  $ row.names:  NULL
## - attributes: List of 2
##  .. $ names: chr [1:3] "virtual" "physical" "row.names"
##  .. $ class: chr "ffdf"
```

1.1.2 Memory mapping with `bigmemory`

The `bigmemory`-package handles data in matrices, and therefore only accepts variables in the same data type. Before importing data via the `bigmemory`-package, we thus have to ensure that all variables in the raw data can be imported in a common type. This example follows the example of the package authors given here.[1]

```
# SET UP ----------------

# load packages
library(bigmemory)
library(biganalytics)

# import the data
flights <- read.big.matrix("../data/flights.csv",
                     type="integer",
                     header=TRUE,
                     backingfile="flights.bin",
                     descriptorfile="flights.desc")
```

Note that, similar to the `ff`-example, `read.big.matrix()` initiates a local file-backing `flights.bin` on disk which is linked to the `flights`-object in RAM. From looking at the imported file, we see that various variable values have been discarded. This is due to the fact that we have forced all variables to be of type `"integer"` when importing the dataset.

```
summary(flights)
```

```
##                      min          max         mean          NAs
## year           2013.000000  2013.000000  2013.000000     0.000000
## month             1.000000    12.000000     6.548510     0.000000
## day               1.000000    31.000000    15.710787     0.000000
## dep_time          1.000000  2400.000000  1349.109947  8255.000000
## sched_dep_time  106.000000  2359.000000  1344.254840     0.000000
## dep_delay       -43.000000  1301.000000    12.639070  8255.000000
## arr_time          1.000000  2400.000000  1502.054999  8713.000000
## sched_arr_time    1.000000  2359.000000  1536.380220     0.000000
## arr_delay       -86.000000  1272.000000     6.895377  9430.000000
## carrier           9.000000     9.000000     9.000000 318316.000000
## flight            1.000000  8500.000000  1971.923620     0.000000
## tailnum                                               336776.000000
## origin                                                336776.000000
## dest                                                  336776.000000
## air_time         20.000000   695.000000   150.686460  9430.000000
## distance         17.000000  4983.000000  1039.912604     0.000000
## hour              1.000000    23.000000    13.180247     0.000000
## minute            0.000000    59.000000    26.230100     0.000000
## time_hour      2013.000000  2014.000000  2013.000261     0.000000
```

2 Cleaning and Transformation

2.1 Typical tasks (independent of data set size)

- Normalize/standardize.

[1] We only use a fraction of the data used in the package vignette example, the full raw data used there can be downloaded here.

- Code additional variables (indicators, strings to categorical, etc.).
- Remove, add covariates.
- Merge data sets.
- Set data types.

2.2 Typical workflow

1. Import raw data.
2. Clean/transform.
3. Store for analysis.
    - Write to file.
    - Write to database.

2.3 Bottlenecks

- RAM:
    - Raw data does not fit into memory.
    - Transformations enlarge RAM allocation (copying).
- Mass Storage: Reading/Writing
- CPU: Parsing (data types)

3 Data Preparation with `ff`

3.1 Set up

The following examples are based on Walkowiak (2016), Chapter 3.

```
## SET UP ------------------------

#Set working directory to the data and airline_id files.
# setwd("materials/code_book/B05396_Ch03_Code")
system("mkdir ffdf")
options(fftempdir = "ffdf")

# load packages
library(ff)
library(ffbase)
library(pryr)

# fix vars
FLIGHTS_DATA <- "../code_book/B05396_Ch03_Code/flights_sep_oct15.txt"
AIRLINES_DATA <- "../code_book/B05396_Ch03_Code/airline_id.csv"
```

3.2 Data import

```
# DATA IMPORT -------------------

# 1. Upload flights_sep_oct15.txt and airline_id.csv files from flat files.
```

```
system.time(flights.ff <- read.table.ffdf(file=FLIGHTS_DATA,
                                          sep=",",
                                          VERBOSE=TRUE,
                                          header=TRUE,
                                          next.rows=100000,
                                          colClasses=NA))
```

```
## read.table.ffdf 1..100000 (100000)  csv-read=0.907sec ffdf-write=0.186sec
## read.table.ffdf 100001..200000 (100000)  csv-read=0.934sec ffdf-write=0.128sec
## read.table.ffdf 200001..300000 (100000)  csv-read=1.06sec ffdf-write=0.126sec
## read.table.ffdf 300001..400000 (100000)  csv-read=0.894sec ffdf-write=0.136sec
## read.table.ffdf 400001..500000 (100000)  csv-read=0.918sec ffdf-write=0.124sec
## read.table.ffdf 500001..600000 (100000)  csv-read=0.907sec ffdf-write=0.114sec
## read.table.ffdf 600001..700000 (100000)  csv-read=0.875sec ffdf-write=0.135sec
## read.table.ffdf 700001..800000 (100000)  csv-read=0.895sec ffdf-write=0.237sec
## read.table.ffdf 800001..900000 (100000)  csv-read=0.914sec ffdf-write=0.109sec
## read.table.ffdf 900001..951111 (51111)  csv-read=0.421sec ffdf-write=0.081sec
##  csv-read=8.725sec  ffdf-write=1.376sec  TOTAL=10.101sec
```

```
##    user  system elapsed
##   9.200   0.809  10.106
```

```
airlines.ff <- read.csv.ffdf(file= AIRLINES_DATA,
                             VERBOSE=TRUE,
                             header=TRUE,
                             next.rows=100000,
                             colClasses=NA)
```

```
## read.table.ffdf 1..1607 (1607)  csv-read=0.012sec ffdf-write=0.008sec
##  csv-read=0.012sec  ffdf-write=0.008sec  TOTAL=0.02sec
```

```
# check memory used
mem_used()
```

```
## 1.33 GB
```

3.3 Comparison with `read.table`

```
##Using read.table()
system.time(flights.table <- read.table(FLIGHTS_DATA,
                                        sep=",",
                                        header=TRUE))
```

```
##    user  system elapsed
##   7.961   0.455   8.432
```

```
gc()
```

```
##             used   (Mb) gc trigger   (Mb) limit (Mb)  max used   (Mb)
## Ncells   1308852   70.0    3390415  181.1         NA   3390415  181.1
## Vcells 157046030 1198.2  266841886 2035.9      16384 266808980 2035.6
```

```
system.time(airlines.table <- read.csv(AIRLINES_DATA,
                                       header = TRUE))
```

```
##    user  system elapsed
##   0.006   0.000   0.006
```

```
# check memory used
mem_used()
```

```
## 1.33 GB
```

3.4 Inspect imported files

```
# 2. Inspect the ffdf objects.
## For flights.ff object:
class(flights.ff)
```

```
## [1] "ffdf"
```

```
dim(flights.ff)
```

```
## [1] 951111     28
```

```
## For airlines.ff object:
class(airlines.ff)
```

```
## [1] "ffdf"
```

```
dim(airlines.ff)
```

```
## [1] 1607    2
```

3.5 Data cleaning and transformation

Goal: merge airline data to flights data

```
# step 1:
## Rename "Code" variable from airlines.ff to "AIRLINE_ID" and "Description" into "AIRLINE_NM".
names(airlines.ff) <- c("AIRLINE_ID", "AIRLINE_NM")
names(airlines.ff)
```

```
## [1] "AIRLINE_ID" "AIRLINE_NM"
```

```
str(airlines.ff[1:20,])
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ AIRLINE_ID: int  19031 19032 19033 19034 19035 19036 19037 19038 19039 19040 ...
##  $ AIRLINE_NM: Factor w/ 1607 levels "40-Mile Air: Q5",..: 945 1025 503 721 64 725 1194 99 1395 276
```

3.6 Data cleaning and transformation

Goal: merge airline data to flights data

```
# merge of ffdf objects
mem_change(flights.data.ff <- merge.ffdf(flights.ff, airlines.ff, by="AIRLINE_ID"))
```

```
## -25 kB
```

```
#The new object is only 551.2 Kb in size
class(flights.data.ff)
```

```
## [1] "ffdf"
```

```
dim(flights.data.ff)
```

```
## [1] 951111     29
```

```
dimnames.ffdf(flights.data.ff)
```

```
## [[1]]
## NULL
##
## [[2]]
##  [1] "YEAR"               "MONTH"              "DAY_OF_MONTH"
##  [4] "DAY_OF_WEEK"        "FL_DATE"            "UNIQUE_CARRIER"
##  [7] "AIRLINE_ID"         "TAIL_NUM"           "FL_NUM"
## [10] "ORIGIN_AIRPORT_ID"  "ORIGIN"             "ORIGIN_CITY_NAME"
## [13] "ORIGIN_STATE_NM"    "ORIGIN_WAC"         "DEST_AIRPORT_ID"
## [16] "DEST"               "DEST_CITY_NAME"     "DEST_STATE_NM"
## [19] "DEST_WAC"           "DEP_TIME"           "DEP_DELAY"
## [22] "ARR_TIME"           "ARR_DELAY"          "CANCELLED"
## [25] "CANCELLATION_CODE"  "DIVERTED"           "AIR_TIME"
## [28] "DISTANCE"           "AIRLINE_NM"
```

3.7 Inspect difference to in-memory operation

```
##For flights.table:
names(airlines.table) <- c("AIRLINE_ID", "AIRLINE_NM")
names(airlines.table)
```

```
## [1] "AIRLINE_ID" "AIRLINE_NM"
```

```
str(airlines.table[1:20,])
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ AIRLINE_ID: int  19031 19032 19033 19034 19035 19036 19037 19038 19039 19040 ...
##  $ AIRLINE_NM: Factor w/ 1607 levels "40-Mile Air: Q5",..: 945 1025 503 721 64 725 1194 99 1395 276
```

```
# check memory usage of merge in RAM
mem_change(flights.data.table <- merge(flights.table,
                                       airlines.table,
                                       by="AIRLINE_ID"))
```

```
## -59 kB
```

```
#The new object is already 105.7 Mb in size
#A rapid spike in RAM use when processing
```

3.8 Subsetting

```
mem_used()
```

```
## 1.33 GB
```

```
# Subset the ffdf object flights.data.ff:
subs1.ff <- subset.ffdf(flights.data.ff, CANCELLED == 1,
                        select = c(FL_DATE, AIRLINE_ID,
                                   ORIGIN_CITY_NAME,
```

```
                                    ORIGIN_STATE_NM,
                                    DEST_CITY_NAME,
                                    DEST_STATE_NM,
                                    CANCELLATION_CODE))

dim(subs1.ff)
```

```
## [1] 4529    7
```

```
mem_used()
```

```
## 1.33 GB
```

3.9 Save to ffdf-files

(For further processing with `ff`)

```
# Save a newly created ffdf object to a data file:

save.ffdf(subs1.ff) #7 files (one for each column) created in the ffdb directory
```

3.10 Load ffdf-files

```
# Loading previously saved ffdf files:
rm(subs1.ff)
gc()
```

```
##             used   (Mb) gc trigger   (Mb) limit (Mb)  max used   (Mb)
## Ncells   1308628   69.9    3390415  181.1         NA   3390415  181.1
## Vcells 157032264 1198.1  266841886 2035.9      16384 266808980 2035.6
```

```
load.ffdf("ffdb")
str(subs1.ff)
```

```
## List of 3
##  $ virtual: 'data.frame':    7 obs. of  7 variables:
##  .. $ VirtualVmode     : chr  "integer" "integer" "integer" "integer" ...
##  .. $ AsIs             : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  .. $ VirtualIsMatrix  : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  .. $ PhysicalIsMatrix : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  .. $ PhysicalElementNo: int  1 2 3 4 5 6 7
##  .. $ PhysicalFirstCol : int  1 1 1 1 1 1 1
##  .. $ PhysicalLastCol  : int  1 1 1 1 1 1 1
##  .. - attr(*, "Dim")= int  4529 7
##  .. - attr(*, "Dimorder")= int  1 2
##  $ physical: List of 7
##  .. $ FL_DATE          : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 4529
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
```

```
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 4529
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:61] "2015-09-01" "2015-09-02" "2015-09-03" "2015-09-04" ...
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ AIRLINE_ID       : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 4529
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 4529
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ ORIGIN_CITY_NAME : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 4529
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 4529
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:305] "Abilene, TX" "Akron, OH" "Albany, GA" "Albany, NY" ...
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ ORIGIN_STATE_NM  : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 4529
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 4529
```

```
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:52] "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ DEST_CITY_NAME   : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 4529
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 4529
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:306] "Abilene, TX" "Akron, OH" "Albany, GA" "Albany, NY" ...
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ DEST_STATE_NM    : list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 4529
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 4529
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:52] "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  .. $ CANCELLATION_CODE: list()
##  ..  ..- attr(*, "physical")=Class 'ff_pointer' <externalptr>
##  ..  .. ..- attr(*, "vmode")= chr "integer"
##  ..  .. ..- attr(*, "maxlength")= int 4529
##  ..  .. ..- attr(*, "pattern")= chr "ffdf"
##  ..  .. ..- attr(*, "filename")= chr "/Users/umatter/Dropbox/Teaching/HSG/BigData/materials/notes/ff
##  ..  .. ..- attr(*, "pagesize")= int 65536
##  ..  .. ..- attr(*, "finalizer")= chr "close"
##  ..  .. ..- attr(*, "finonexit")= logi TRUE
##  ..  .. ..- attr(*, "readonly")= logi FALSE
##  ..  .. ..- attr(*, "caching")= chr "mmnoflush"
##  ..  ..- attr(*, "virtual")= list()
##  ..  .. ..- attr(*, "Length")= int 4529
##  ..  .. ..- attr(*, "Symmetric")= logi FALSE
##  ..  .. ..- attr(*, "Levels")= chr [1:4] "" "A" "B" "C"
##  ..  .. ..- attr(*, "ramclass")= chr "factor"
```

```
##  .. .. - attr(*, "class") =  chr [1:2] "ff_vector" "ff"
##  $ row.names:  NULL
## - attributes: List of 2
##  .. $ names: chr [1:2] "virtual" "physical"
##  .. $ class: chr "ffdf"
```

```
dim(subs1.ff)
```

```
## [1] 4529    7
```

```
dimnames(subs1.ff)
```

```
## [[1]]
## NULL
##
## [[2]]
## [1] "FL_DATE"           "AIRLINE_ID"        "ORIGIN_CITY_NAME"
## [4] "ORIGIN_STATE_NM"   "DEST_CITY_NAME"    "DEST_STATE_NM"
## [7] "CANCELLATION_CODE"
```

3.11 Export to CSV

```
#  Export subs1.ff into CSV and TXT files:
write.csv.ffdf(subs1.ff, "subset1.csv")
```

References

Walkowiak, Simkon. 2016. *Big Data Analytics with R*. Birmingham, UK: PACKT Publishing.