RESEARCH INSTITUTE
NATURE AND FOREST

Flanders
State of
the Art

git2rdata

Storing Dataframes in a Plain Text Format Suitable for Version Control

useR!2019, Toulouse

Thierry Onkelinx

Research Institute for Nature and Forest (INBO)

git2rdata

www.INBO.be

Requirements

1. open, plain text format
2. read data = stored data
3. compact storage on disk
4. meaningful history
5. integrates with analysis

Store Data in Unambiguous Format

Data

- tab separated file (.tsv)

Metadata

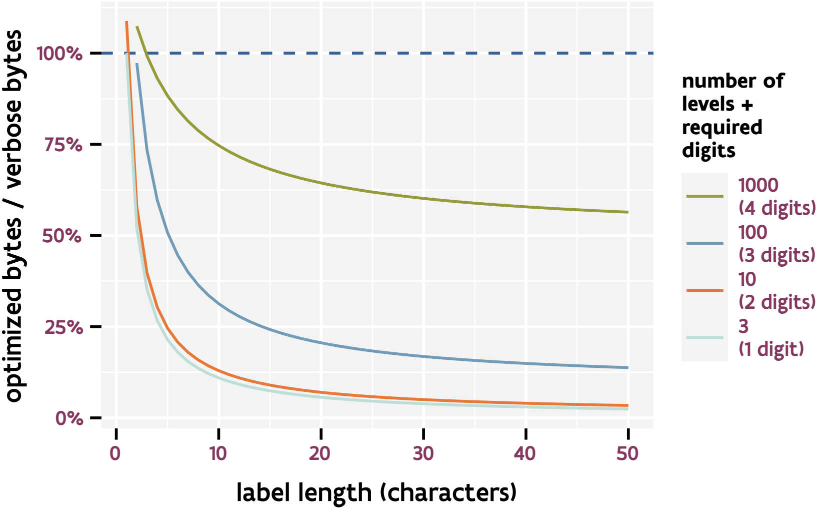
- YAML file (.yml)
- class + format

Optimise File Storage

method	relative size	file size (KiB)
saveRDS()	12%	300
write_vc(), *optimized*	**64%**	1700
write_vc(), *verbose*	**91%**	2500
write.table()	100%	2700

- example data set
  - airbag dataset from DAAG package
  - 26K observations, 16 variables (7 integer, 5 factors, 3 logical, 1 numeric)
- *verbose*
  - human readable format
  - larger data file
- *optimized*
  - maximal use of metadata to minimize data file
  - less human readable

Optimise Storage of Factors



Flanders
State of the Art

Usage on a File System

```r
my_project <- "~/project_dir"
```

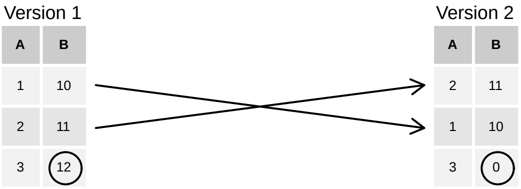
```r
library(git2rdata)
write_vc(iris, file = "my_data", root = my_project)
```

```
## 18f7faabae2373f138476782bc5537cee61d3b9a
##                             "my_data.tsv"
## a21c64fa1d6cb4d7014f4fd33571e77c62556e59
##                             "my_data.yml"
```

```r
stored <- read_vc("my_data", root = my_project)
all.equal(iris, stored, check.attributes = FALSE)
```

```
## [1] TRUE
```

Optimise Storage under Version Control



Version 1

A	B
1	10
2	11
3	12

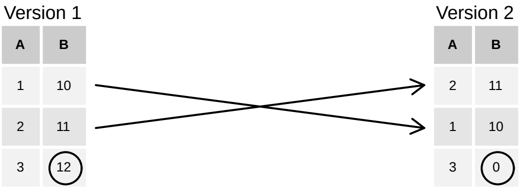
Version 2

A	B
2	11
1	10
3	0

Flanders
State of the Art

git2rdata

Optimise Storage under Version Control



Version 1

A	B
1	10
2	11
3	12

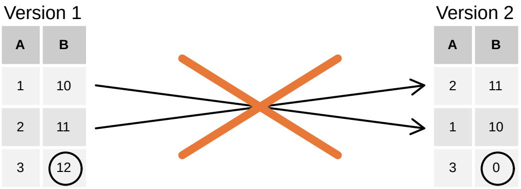
Version 2

A	B
2	11
1	10
3	0

diff on unsorted data

	A	B
+	2	11
	1	10
-	2	11
-	3	12
+	3	0

Optimise Storage under Version Control



Version 1

A	B
1	10
2	11
3	12

Version 2

A	B
2	11
1	10
3	0

diff on unsorted data

	A	B
+	2	11
	1	10
-	2	11
-	3	12
+	3	0

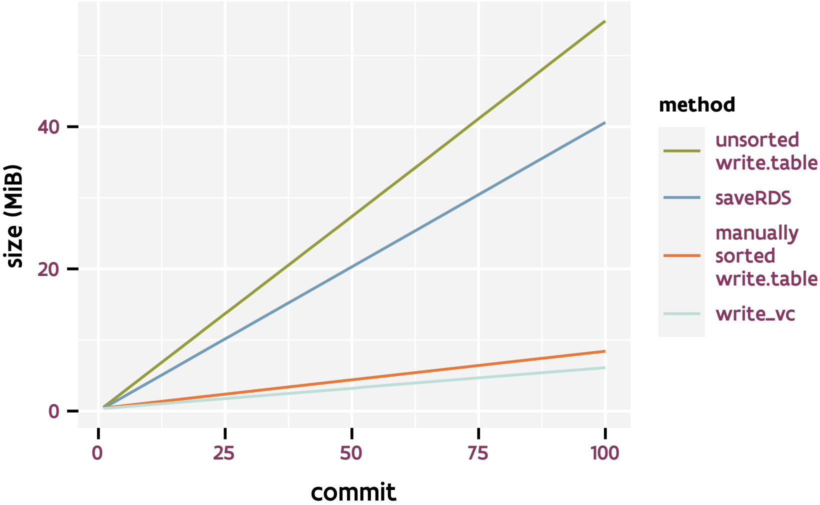
diff on sorted data

	A	B
	1	10
	2	11
-	3	12
+	3	0

Flanders
State of the Art

git2rdata

Git History Size



Git Size Recommendations

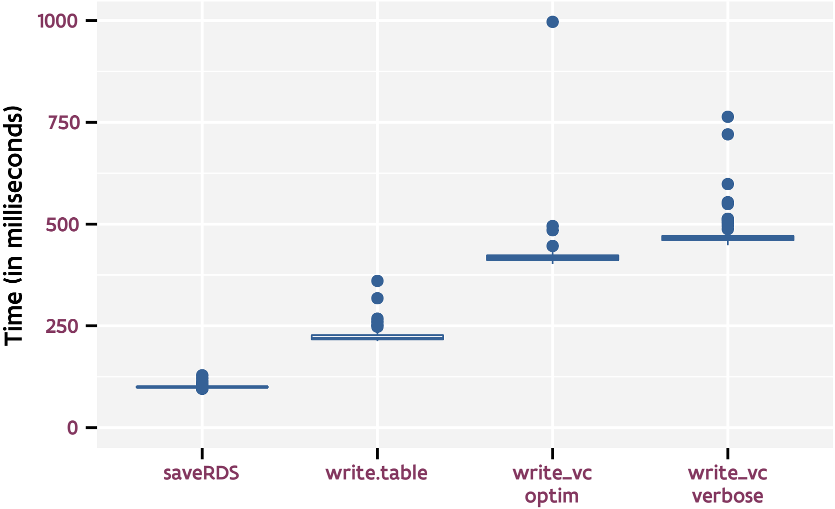
Keep File under 100 MiB

airbag data	optimised	verbose
bytes per observation	68 B	97 B
max. observations	1.5M	1M

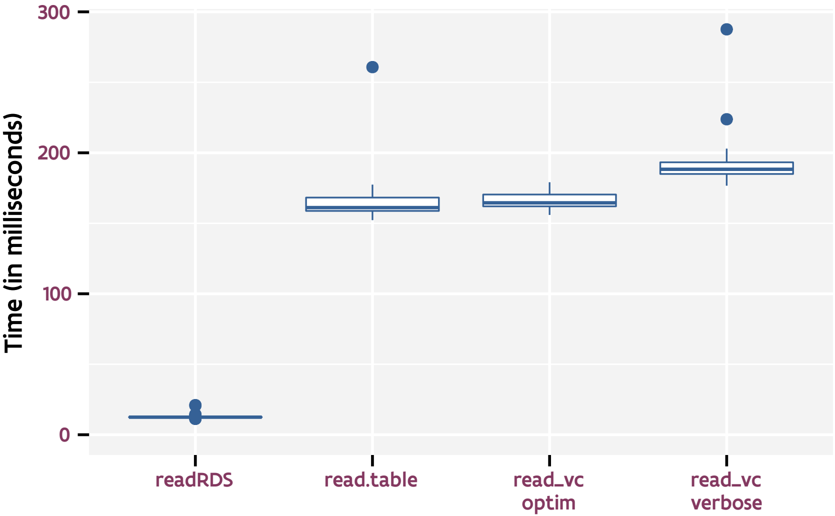
Keep History under 1 GiB

airbag data	optimised	verbose
delta 90% random subset	60 KiB	100 KiB
max. commits	17k	10k

How Fast is Writing?



How Fast is Reading?



Flanders
State of the Art

git2rdata

Build-in Safeguards

```
my_repo <- repository(my_project) # git2r repo object
mtcars <- rownames_to_column(mtcars, "model_make")

hash <- write_vc(mtcars, file = "cars/mt", root = my_repo, stage = TRUE)
```

```
## Warning: No sorting applied.
## Sorting is strongly recommended in combination with version control.
```

```
hash <- write_vc(mtcars, file = "cars/mt", root = my_repo, stage = TRUE,
                 sorting = "model_make")
```

```
## Error: The data was not overwritten because of the issues below.
## See vignette('version_control', package = 'git2rdata') for more information.
##
## - The sorting variables changed.
##     - Sorting for the new data: 'model_make'.
##     - Sorting for the old data: .
```

Overriding Build-in Safeguards

```r
hash <- write_vc(mtcars, file = "cars/mt", root = my_repo, stage = TRUE,
                 sorting = "model_make", strict = FALSE)
```

```
## Warning: Changes in the metadata may lead to unnecessarily large diffs.
## See vignette('version_control', package = 'git2rdata') for more information.
##
## - The sorting variables changed.
##     - Sorting for the new data: 'model_make'.
##     - Sorting for the old data: .
```

```r
commit(my_repo, "initial version of mtcars")
```

```
## [5fc4137] 2019-07-10: initial version of mtcars
```

Analysis Workflow with Reproducible Data

What was the latest update of the git2rdata object?

```
recent_commit("cars/mt", root = my_repo, data = TRUE)
```

```
##                                      commit           author                when
## 1 5fc4137d485c11e9523e4ba9fc795bc6197888aa Thierry Onkelinx 2019-07-10 05:45:33
```

Store repo and commit metadata alongside the analysis

```
analysis <- function(ds_name, repo) {
  ds <- read_vc(ds_name, repo)
  list(
    repository = git2r::remote_url(repo),
    commit = recent_commit(ds_name, repo, data = TRUE),
    model = lm(mpg ~ disp, data = ds)
  )
}
```

Getting Started

`https://ropensci.github.io/git2rdata` or CRAN

- installation, vignettes and documentation

Thanks to **Brodie Gaslam** and **Joyce Robbins** for their reviews on rOpenSci

Flanders
State of the Art

git2rdata