Importing & Exporting Data with R

Aditya Dube

Monday, May 15, 2023

Loading Packages

```
library(data.table)
library(tidyverse)
library(knitr)
library(arrow)
library(bench)
library(ggbeeswarm)
library(haven)
```

Import the possum.csv data directly from the URL

```
possums <- read.csv("https://raw.githubusercontent.com/dilernia/STA418-518/main/Data/possum.csv")

# Printing first 6 columns and first 5 rows from the dataset
possums %>% dplyr::select(1:6) %>%
  slice_head(n = 5) %>% kable()
```

case	site	sex	age	head_length_mm	skull_width_mm
1	Cambarville	Male	8	94.1	60.4
2	Cambarville	Female	6	92.5	57.6
3	Cambarville	Female	6	94.0	60.0
4	Cambarville	Female	6	93.2	57.1
5	Cambarville	Female	2	91.5	56.3

Importing using the import wizard

```
possums_1 <- read_csv("possum.csv")
```

```
## Rows: 104 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): site, sex, region
## dbl (11): case, age, head_length_mm, skull_width_mm, total_length_cm, tail_l...
```

```
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

Using read_csv

```
possums <- read_csv("https://raw.githubusercontent.com/dilernia/STA418-518/main/Data/possum.csv")
```

```
## Rows: 104 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): site, sex, region
## dbl (11): case, age, head_length_mm, skull_width_mm, total_length_cm, tail_l...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
# Printing first 6 columns and first 5 rows from the dataset
possums %>% dplyr::select(1:6) %>%
  slice_head(n = 5) %>% kable()
```

case	site	sex	age	head_length_mm	skull_width_mm
1	Cambarville	Male	8	94.1	60.4
2	Cambarville	Female	6	92.5	57.6
3	Cambarville	Female	6	94.0	60.0
4	Cambarville	Female	6	93.2	57.1
5	Cambarville	Female	2	91.5	56.3

Using read_parquet

```
possums <- read_parquet("possum.parquet")

# Printing first 8 columns and first 5 rows from the dataset
possums %>% dplyr::select(1:8) %>%
  slice_head(n = 5) %>% kable()
```

case	site	sex	age	head_length_mm	skull_width_mm	total_length_cm	tail_length_cm
1	Cambarville	Male	8	94.1	60.4	89.0	36.0
2	Cambarville	Female	6	92.5	57.6	91.5	36.5
3	Cambarville	Female	6	94.0	60.0	95.5	39.0
4	Cambarville	Female	6	93.2	57.1	92.0	38.0
5	Cambarville	Female	2	91.5	56.3	85.5	36.0

Using fread()

```r
possum_fread <- fread("https://raw.githubusercontent.com/dilernia/STA418-518/main/Data/possum.csv")

# Printing first 6 columns and first 5 rows from the dataset
possum_fread %>% dplyr::select(1:6) %>%
  slice_head(n = 5) %>% kable()
```

case	site	sex	age	head_length_mm	skull_width_mm
1	Cambarville	Male	8	94.1	60.4
2	Cambarville	Female	6	92.5	57.6
3	Cambarville	Female	6	94.0	60.0
4	Cambarville	Female	6	93.2	57.1
5	Cambarville	Female	2	91.5	56.3

Comparing read speeds

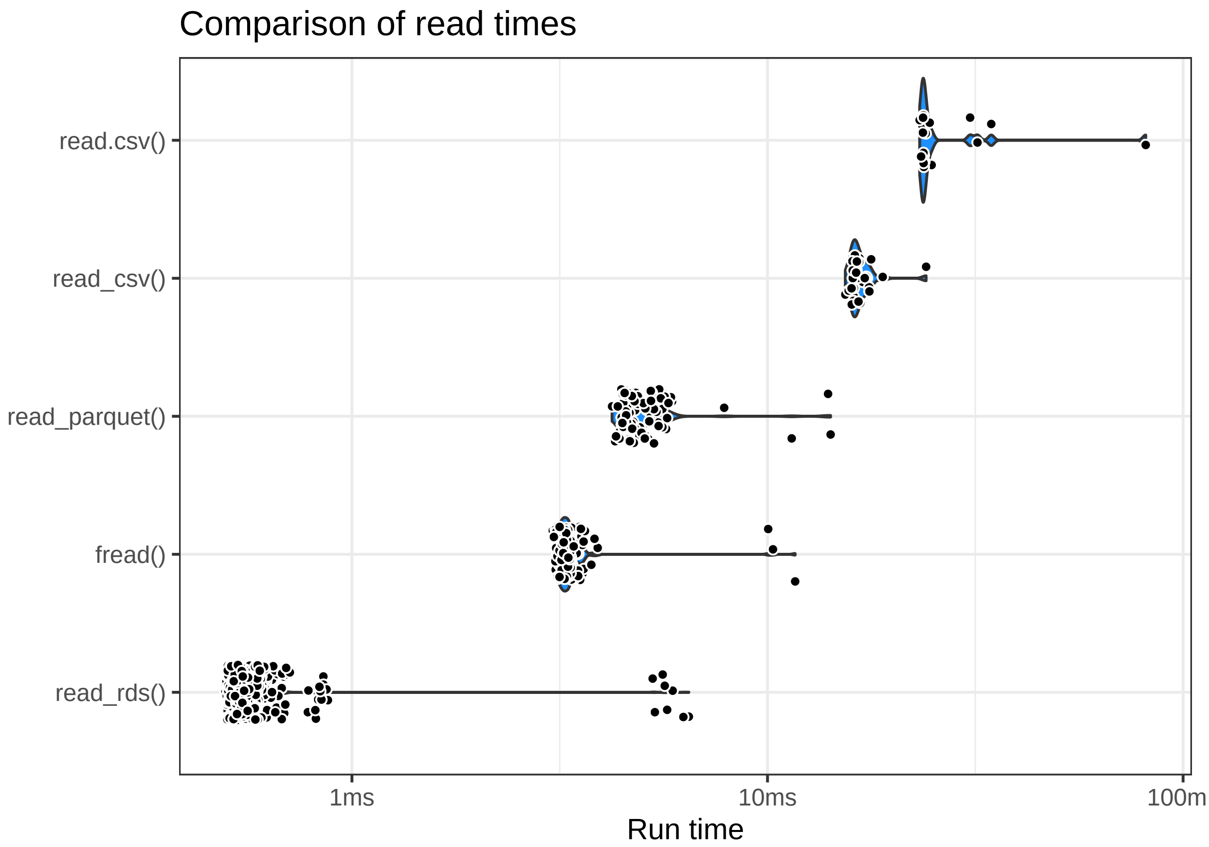
```r
# Generating 'big data'
set.seed(1994)
x <- runif(5e4)
y <- runif(5e4)
x[sample(5e4, 5e3)] <- NA
y[sample(5e4, 5e3)] <- NA
bigData <- as.data.frame(x = x, y = y)
# Saving as CSV file w/ data.table
fwrite(bigData, "bigData.csv")
# Saving as parquet file
write_parquet(bigData, "bigData.parquet")
# Saving as RDS file
write_rds(bigData, "bigData.rds")
```

```r
# Comparing run times
readBmResult <- mark(read.csv("bigData.csv"), read_csv("bigData.csv",
                                                       show_col_types = FALSE),
                     fread("bigData.csv"), read_rds("bigData.rds"),
                     read_parquet("bigData.parquet", as_tibble = TRUE),
                     check = FALSE, min_iterations = 5)
ggObj <- plot(readBmResult)
importTimes <- ggObj$data %>% mutate(expression =
                                       paste0(map_chr(str_split(expression, pattern = "[(]"), 1), "()")
# Printing table
importTimes %>% arrange(desc(median)) %>%
  select(expression:mem_alloc) %>% distinct() %>% knitr::kable()
```

expression	min	median	itr/sec	mem_alloc
read.csv()	23.23ms	23.76ms	39.60310	1.7MB
read_csv()	15.39ms	16.38ms	60.34031	359.5KB
read_parquet()	4.23ms	4.77ms	202.82383	204.5KB
fread()	3.04ms	3.27ms	303.86058	795.7KB

expression	min	median	itr/sec	mem_alloc
read_rds()	495.62us	551.76us	1786.85366	395.8KB

```
# Creating violin plots
importTimes %>% ggplot(aes(x = time, y = fct_reorder(expression, time))) +
  geom_violin(fill = "dodgerblue") +
  geom_jitter(
    height = 0.2,
    pch = 21,
    fill = "black",
    color = "white"
  ) +
  labs(title = "Comparison of read times", y = "", x = "Run time") +
  theme_bw()
```



Reproduce the table and violin plots for write function

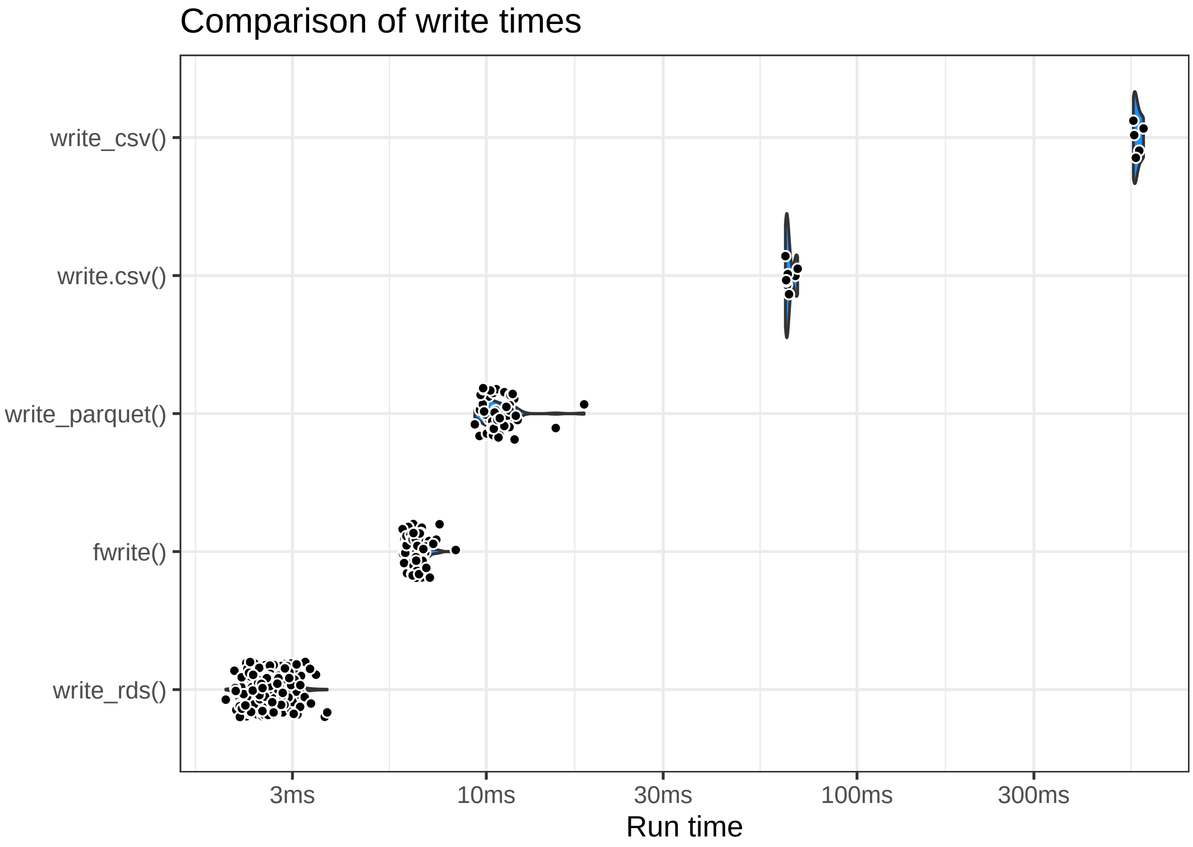
```
# Comparing run times
writeBmResult <- mark(write.csv(bigData, "bigData.csv"), write_csv(bigData, "bigData.csv"),fwrite(bigDa

ggObj <- plot(writeBmResult)

exportTimes <- ggObj$data %>% mutate(expression =
                    paste0(map_chr(str_split(expression, pattern = "[(]"), 1), "()")
# Printing table
exportTimes %>% arrange(desc(median)) %>%
  select(expression:mem_alloc) %>% distinct() %>% knitr::kable()
```

expression	min	median	itr/sec	mem_alloc
write_csv()	556.43ms	565.08ms	1.754052	102.63KB
write.csv()	64.09ms	64.91ms	15.220177	1.51MB
write_parquet()	9.31ms	10.57ms	91.340077	15.39KB
fwrite()	5.95ms	6.46ms	153.954858	0B
write_rds()	1.98ms	2.56ms	384.425784	8.63KB

```
# Creating violin plots
exportTimes %>% ggplot(aes(x = time, y = fct_reorder(expression, time))) +
  geom_violin(fill = "dodgerblue") +
  geom_jitter(
    height = 0.2,
    pch = 21,
    fill = "black",
    color = "white"
  ) +
  labs(title = "Comparison of write times", y = "", x = "Run time") +
  theme_bw()
```


Comparison of write times
write_csv()
write.csv()
write_parquet()
fwrite()
write_rds()
3ms
10ms
30ms
100ms
300ms
Run time