Tibbles vs. data.frame

Lauren Kapraun

09/17/18

Before starting, I installed the following: **tibble** packages and the **tibble, tidyverse, tidyr, magrittr** libraries.

The two main fifferences in using a tibble versus the `data.frame` is printing and subsetting.

Printing

When using tibbles, the refined print method only displays the first 10 rows by default.

```
tb = tibble(
  a = lubridate::now() + runif(1e3) * 86400,
  b = lubridate::today() + runif(1e3) * 30,
  c = 1:1e3,
  d = runif(1e3),
  e = sample(letters, 1e3, replace = TRUE)
)
```

Display Options

One option you can use to display more than the first 10 rows is to use the `print(n)` function, where "n" is the number of rows you want to display. If some of the columns are hidden due to to size, you can also use `width=Inf` to show all of the columns in the table.

```
tb %>%
  print(n = 15, width = Inf)
```

```
## # A tibble: 1,000 x 5
##    a                   b              c      d e
##    <dttm>              <date>     <int>  <dbl> <chr>
##  1 2018-09-13 14:53:38 2018-10-07     1 0.878  w
##  2 2018-09-13 20:18:49 2018-09-13     2 0.844  b
##  3 2018-09-13 19:22:06 2018-09-19     3 0.627  u
##  4 2018-09-13 09:01:34 2018-09-21     4 0.494  k
##  5 2018-09-13 13:55:20 2018-09-18     5 0.0755 g
##  6 2018-09-13 12:48:52 2018-09-26     6 0.220  p
##  7 2018-09-13 06:00:16 2018-10-01     7 0.547  z
##  8 2018-09-13 11:42:36 2018-10-02     8 0.898  z
##  9 2018-09-13 22:45:40 2018-09-29     9 0.105  p
## 10 2018-09-13 18:52:19 2018-10-03    10 0.815  p
## 11 2018-09-13 21:31:59 2018-09-25    11 0.469  u
## 12 2018-09-13 05:16:05 2018-09-13    12 0.146  g
## 13 2018-09-13 11:01:00 2018-09-12    13 0.855  x
## 14 2018-09-13 00:42:17 2018-10-03    14 0.877  a
## 15 2018-09-13 20:28:16 2018-09-30    15 0.239  u
## # ... with 985 more rows
```

Display Min & Max

To show a minimum or a maximum number of rows use: `options(tibble.print_max = n, tribble.print_min = m)`. In this instance, if there are more than "m" rows, it will display all the rows up to "n" rows.

```
options(tibble.print_max = 30, tibble.print_min = 20)
  tb
```

Disply All Rows

To always display all rows, use: `options(dplyr.print_min = Inf)`.

```
options(dplyr.print_min = Inf)
  tb
```

Disply All Columns

To show all columns by default, use: `options(tribble.width = Inf)`

```
options(n = 15, tribble.width = Inf)
  tb
```

Disply All in New Tab

You can also use RStudio's built-in data viewer to display a scrollable view of the dataset using `View()`.

Final Print Options

To view all of the package options, run `package?tibble`

Subsetting

To pull from a single vairable, you can use `[[` to extract by name or position, or you can use `$` to extract only by name.

Tibbles are more strict than data.frames, as they do not allow for partial matching.

```
df = tibble(
  x = runif(5),
  y = runif(5)
)

# Extract by Name
df$x
```

```
## [1] 0.76553542 0.85817816 0.01264303 0.88478504 0.36337227
```

```
df[["x"]]
```

```
## [1] 0.76553542 0.85817816 0.01264303 0.88478504 0.36337227
```

```
# Extract by Position
df[[1]]
```

```
## [1] 0.76553542 0.85817816 0.01264303 0.88478504 0.36337227
```

```
# To Use in a Pipe, Use a Special Placeholder
df %>% .$x
```

```
## [1] 0.76553542 0.85817816 0.01264303 0.88478504 0.36337227
```

```
df %>% .[["x"]]
```

```
## [1] 0.76553542 0.85817816 0.01264303 0.88478504 0.36337227
```

Interacting with Older Code

If you are trying to use an older function that doesn't seem to be working, use `as.data.frame()` to convert the tibble to a data.frame.

```
class(as.data.frame(tb))
```