

Rsquared Academy



Introduction to tibbles

Agenda

- what are tibbles?
- how are tibbles different from data frames?
- how to create tibbles?
- how to manipulate tibbles?

What are tibbles?

What are tibbles?

A **tibble**, or `tbl_df`, is a modern reimagining of the `data.frame`, keeping what time has proven to be effective, and throwing out what is not. Tibbles are `data.frames` that are lazy and surly: they do less (i.e. they don't change variable names or types, and don't do partial matching) and complain more (e.g. when a variable does not exist). This forces you to confront problems earlier, typically leading to cleaner, more expressive code. Tibbles also have an enhanced `print method()` which makes them easier to use with large datasets containing complex objects.

Source: <http://tibble.tidyverse.org/>

Creating tibbles

```
tibble(x = letters,  
       y = 1:26,  
       z = sample(100, 26))
```

```
## # A tibble: 26 x 3  
##       x         y         z  
##   <chr> <int> <int>  
## 1 a             1      66  
## 2 b             2      63  
## 3 c             3      35  
  
## 4 d             4      54  
## 5 e             5      13  
## 6 f             6       5  
## 7 g             7      39  
## 8 h             8       4  
## 9 i             9      25  
## 10 j            10      14  
## # ... with 16 more rows
```

tibble Features

- never changes input's types
- never adjusts variable names
- never prints all rows
- never recycles vector of length greater than 1


```
tibble(x = letters,  
       y = 1:26,  
       z = sample(100, 26))
```

```
## # A tibble: 26 x 3  
##       x         y         z  
##   <chr> <int> <int>  
## 1 a         1      39  
## 2 b         2      87  
## 3 c         3      70  
  
## 4 d         4       5  
## 5 e         5     91  
## 6 f         6     19  
## 7 g         7     54  
## 8 h         8     29  
## 9 i         9     48  
## 10 j        10     13  
## # ... with 16 more rows
```

```
data <- data.frame(x = letters, y = 1:26, z = sample(100, 26))  
str(data)
```

```
## 'data.frame':    26 obs. of  3 variables:  
##  $ x: Factor w/ 26 levels "a","b","c","d",...: 1 2 3 4 5 6 7 8 9 10 ..  
##  $ y: int  1 2 3 4 5 6 7 8 9 10 ...  
##  $ z: int  16 42 94 40 68 29 13 50 34 79 ...
```

```
names(data.frame(`order value` = 10))
```

```
## [1] "order.value"
```

```
names(tibble(`order value` = 10))
```

```
## [1] "order value"
```

```
x <- 1:100  
y <- letters[1]  
z <- sample(c(TRUE, FALSE), 100, replace = TRUE)  
tibble(x, y, z)
```

```
## # A tibble: 100 x 3  
##       x y      z  
##   <int> <chr> <lgl>  
## 1     1 a    FALSE  
## 2     2 a     TRUE  
## 3     3 a     TRUE  
## 4     4 a     TRUE  
## 5     5 a     TRUE  
## 6     6 a     TRUE  
## 7     7 a    FALSE  
## 8     8 a    FALSE  
## 9     9 a    FALSE  
## 10    10 a    FALSE  
## # ... with 90 more rows
```

```
x <- 1:100  
y <- letters  
z <- sample(c(TRUE, FALSE), 100, replace = TRUE)  
tibble(x, y, z)  
Error in overscope_eval_next(overscope, expr) : object 'y' not found
```

enframe

```
browsers <- c('chrome', 'safari', 'firefox', 'edge')
enframe(browsers)
```

```
## # A tibble: 4 x 2
##   name value
##   <int> <chr>
## 1     1 chrome
## 2     2 safari
## 3     3 firefox
## 4     4 edge
```

```
browsers <- c(chrome = 40, firefox = 20, edge = 30, safari = 10)
enframe(browsers)
```

```
## # A tibble: 4 x 2
##   name      value
##   <chr>    <dbl>
## 1 chrome      40
## 2 firefox     20
## 3 edge       30
## 4 safari     10
```


tribble

Another way to create tibbles is using `tribble()`:

- it is short for transposed tibbles
- it is customized for data entry in code
- column names start with ~
- and values are separated by commas

```
tribble(  
  ~x, ~y, ~z,  
  #--|--|-----  
  1, TRUE, 'a',  
  2, FALSE, 'b'  
)
```

```
## # A tibble: 2 x 3  
##       x y     z  
##   <dbl> <lgl> <chr>  
## 1     1  TRUE  a  
## 2     2 FALSE  b
```

Names of the columns in tibbles need not be valid R variable names. They can contain unusual characters like a space or a smiley but must be enclosed in ticks.

```
tibble(  
  ` ` = 'space',  
  `2` = 'integer',  
  `:)` = 'smiley'  
)
```

```
## # A tibble: 1 x 3  
##   ` ` `2` `:)`  
##   <chr> <chr> <chr>  
## 1 space integer smiley
```

Manipulate tibble

```
browsers <- enframe(c(chrome = 40, firefox = 20, edge = 30))  
browsers
```

```
## # A tibble: 3 x 2  
##   name      value  
##   <chr>    <dbl>  
## 1 chrome      40  
## 2 firefox     20  
## 3 edge        30
```

Add Rows

```
add_row(browsers, name = 'safari', value = 10)
```

```
## # A tibble: 4 x 2
##   name      value
##   <chr>    <dbl>
## 1 chrome     40
## 2 firefox    20
## 3 edge       30
## 4 safari     10
```

Add Rows

```
add_row(browsers, name = 'safari', value = 10, .before = 2)
```

```
## # A tibble: 4 x 2
##   name      value
##   <chr>    <dbl>
## 1 chrome     40
## 2 safari     10
## 3 firefox    20
## 4 edge       30
```


Add Column

```
browsers <- enframe(c(chrome = 40, firefox = 20, edge = 30, safari = 10)  
add_column(browsers, visits = c(4000, 2000, 3000, 1000))
```

```
## # A tibble: 4 x 3  
##   name      value visits  
##   <chr>    <dbl>  <dbl>  
## 1 chrome      40    4000  
## 2 firefox     20    2000  
## 3 edge        30    3000  
## 4 safari      10    1000
```

Remove Rownames

```
remove_rownames(mtcars)
```

##	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## 1	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
## 2	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
## 3	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
## 4	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
## 5	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
## 6	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
## 7	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
## 8	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
## 9	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
## 10	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
## 11	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
## 12	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
## 13	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
## 14	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
## 15	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
## 16	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4

```
head(rownames_to_column(mtcars))
```

##	rowname	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	car
## 1	Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	
## 2	Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	
## 3	Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	
## 4	Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	
## 5	Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	
## 6	Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	

Column to Rownames

```
mtcars_tbl <- rownames_to_column(mtcars)
column_to_rownames(mtcars_tbl)
```

##	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	ca
## Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	
## Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	
## Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	
## Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	
## Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	
## Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	
## Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	
## Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	
## Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	
## Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	
## Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	
## Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	
## Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	
## Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	
## Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	
## Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	

```
glimpse(mtcars)
```

```
## Observations: 32
## Variables: 11
## $ mpg    <dbl> 21.0, 21.0, 22.8, 21.4, 18.7, 18.1, 14.3, 24.4, 22.8, 19
## $ cyl    <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8, 8, 8, 8, 8, 8, 4, 4,
## $ disp   <dbl> 160.0, 160.0, 108.0, 258.0, 360.0, 225.0, 360.0, 146.7,
## $ hp     <dbl> 110, 110, 93, 110, 175, 105, 245, 62, 95, 123, 123, 180,
## $ drat   <dbl> 3.90, 3.90, 3.85, 3.08, 3.15, 2.76, 3.21, 3.69, 3.92, 3.
## $ wt     <dbl> 2.620, 2.875, 2.320, 3.215, 3.440, 3.460, 3.570, 3.190,
## $ qsec   <dbl> 16.46, 17.02, 18.61, 19.44, 17.02, 20.22, 15.84, 20.00,
## $ vs     <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1,
## $ am     <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
## $ gear   <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 4, 4,
## $ carb   <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4, 3, 3, 3, 4, 4, 4, 1, 2,
```

```
is_tibble(mtcars)
```

```
## [1] FALSE
```

```
is_tibble(as_tibble(mtcars))
```

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

Rownames

```
has_rownames(mtcars)
```

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

Check Column

```
has_name(mtcars, 'cyl')
```

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

```
has_name(mtcars, 'gears')
```

```
## [1] FALSE
```


Summary

- use `tibble()` to create tibbles
- use `as_tibble()` to coerce other objects to tibble
- use `enframe()` to coerce vector to tibble
- use `tribble()` to create tibble using data entry

Summary

- use `add_row()` to add a new row
- use `add_column()` to add a new column
- use `remove_rownames()` to remove rownames from data
- use `rownames_to_column()` to coerce rowname to first column
- use `column_to_rownames()` to coerce first column to rownames

Summary

- use `is_tibble()` to test if an object is a tibble
- use `has_rownames()` to check whether a data set has rownames
- use `has_name()` to check if tibble has a specific column
- use `glimpse()` to get an overview of data



Thank You

For more information please visit our website
www.rsquaredacademy.com