Chapter7

Ayush Kumar Shah

9/23/2020

Chapter 7 Tibbles

Tibbles are one of the unifying features of the tidyverse

Coercing a dataframe to a tibble:

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------------

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------------------------------ ti
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
class(iris)
```

```
## [1] "data.frame"
```

```
as_tibble(iris)
```

```
## # A tibble: 150 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           <dbl>       <dbl>        <dbl>       <dbl> <fct>
##  1          5.1         3.5          1.4         0.2 setosa
##  2          4.9         3            1.4         0.2 setosa
##  3          4.7         3.2          1.3         0.2 setosa
##  4          4.6         3.1          1.5         0.2 setosa
##  5          5           3.6          1.4         0.2 setosa
##  6          5.4         3.9          1.7         0.4 setosa
##  7          4.6         3.4          1.4         0.3 setosa
##  8          5           3.4          1.5         0.2 setosa
##  9          4.4         2.9          1.4         0.2 setosa
## 10          4.9         3.1          1.5         0.1 setosa
## # ... with 140 more rows
```

Creating tibble

```
tibble(
x = 1:5,
y = 1,
z = x ^ 2 + y
)
```

```
## # A tibble: 5 x 3
##       x     y     z
##   <int> <dbl> <dbl>
## 1     1     1     2
## 2     2     1     5
## 3     3     1    10
## 4     4     1    17
## 5     5     1    26
```

Characters not supported by R also can be column names in tibble

```
tb <- tibble(
`:)` = "smile",
` ` = "space",
`2000` = "number"
)
tb
```

```
## # A tibble: 1 x 3
##   ‘:)‘  ‘ ‘   ‘2000‘
##   <chr> <chr> <chr>
## 1 smile space number
```

Creating tiblle Row-wise

```
tribble(
~x, ~y, ~z,
#--/--/----
"a", 2, 3.6,
"b", 1, 8.5
)
```

```
## # A tibble: 2 x 3
##   x         y     z
##   <chr> <dbl> <dbl>
## 1 a         2   3.6
## 2 b         1   8.5
```

Tibble vs dataframe

Printing

- only the first 10 rows, and all the columns that fit on screen
- each column reports its type

To print more,

```
nycflights13::flights %>%
print(n = 10, width = Inf)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
##    arr_delay carrier flight tailnum origin dest  air_time distance  hour minute
##        <dbl> <chr>    <int> <chr>   <chr>  <chr>    <dbl>    <dbl> <dbl>  <dbl>
##  1        11 UA        1545 N14228  EWR    IAH        227     1400     5     15
##  2        20 UA        1714 N24211  LGA    IAH        227     1416     5     29
##  3        33 AA        1141 N619AA  JFK    MIA        160     1089     5     40
##  4       -18 B6         725 N804JB  JFK    BQN        183     1576     5     45
##  5       -25 DL         461 N668DN  LGA    ATL        116      762     6      0
##  6        12 UA        1696 N39463  EWR    ORD        150      719     5     58
##  7        19 B6         507 N516JB  EWR    FLL        158     1065     6      0
##  8       -14 EV        5708 N829AS  LGA    IAD         53      229     6      0
##  9        -8 B6          79 N593JB  JFK    MCO        140      944     6      0
## 10         8 AA         301 N3ALAA  LGA    ORD        138      733     6      0
##    time_hour
##    <dttm>
##  1 2013-01-01 05:00:00
##  2 2013-01-01 05:00:00
##  3 2013-01-01 05:00:00
##  4 2013-01-01 05:00:00
##  5 2013-01-01 06:00:00
##  6 2013-01-01 05:00:00
##  7 2013-01-01 06:00:00
##  8 2013-01-01 06:00:00
##  9 2013-01-01 06:00:00
## 10 2013-01-01 06:00:00
## # ... with 336,766 more rows
```

- `options(tibble.print_max = n, tibble.print_min = m)`: if more than m rows, print only n rows. Use

- `options(dplyr.print_min = Inf)` to always show all rows.
- Use `options(tibble.width = Inf)` to always print all columns, regardless of the width of the screen.

```
?tibble
```

```
## Help on topic 'tibble' was found in the following packages:
##
##   Package               Library
##   dplyr                 /Library/Frameworks/R.framework/Versions/4.0/Resources/library
##   tibble                /Library/Frameworks/R.framework/Versions/4.0/Resources/library
##   tidyr                 /Library/Frameworks/R.framework/Versions/4.0/Resources/library
##
##
## Using the first match ...
```

Subsetting

```
df <- tibble(
x = runif(5),
y = rnorm(5)
)
options(digits=2)
```

```
df$x
```

```
## [1] 0.63 0.29 0.60 0.52 0.59
```

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

```
## [1] 0.63 0.29 0.60 0.52 0.59
```

```
df[[1]]
```

```
## [1] 0.63 0.29 0.60 0.52 0.59
```

Using pipe

```
df %>% .$x
```

```
## [1] 0.63 0.29 0.60 0.52 0.59
```

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

```
## [1] 0.63 0.29 0.60 0.52 0.59
```

To use functions not supported for tibbles: convert to dataframe

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

```
## [1] "data.frame"
```