

Lecture Assignment 10

Taiki Yamashita

2024-05-08

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.0      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

— 10.5 Exercise 1 —

How can you tell if an object is a tibble? (Hint: try printing mtcars, which is a regular data frame).

When we print mtcars, it prints all of the columns.

mtcars

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160.0 110 3.90 2.620 16.46 0  1    4    4
## Mazda RX4 Wag  21.0   6  160.0 110 3.90 2.875 17.02 0  1    4    4
## Datsun 710     22.8   4  108.0  93 3.85 2.320 18.61 1  1    4    1
## Hornet 4 Drive  21.4   6  258.0 110 3.08 3.215 19.44 1  0    3    1
## Hornet Sportabout 18.7   8  360.0 175 3.15 3.440 17.02 0  0    3    2
## Valiant        18.1   6  225.0 105 2.76 3.460 20.22 1  0    3    1
## Duster 360     14.3   8  360.0 245 3.21 3.570 15.84 0  0    3    4
## Merc 240D      24.4   4  146.7  62 3.69 3.190 20.00 1  0    4    2
## Merc 230       22.8   4  140.8  95 3.92 3.150 22.90 1  0    4    2
## Merc 280       19.2   6  167.6 123 3.92 3.440 18.30 1  0    4    4
## Merc 280C      17.8   6  167.6 123 3.92 3.440 18.90 1  0    4    4
## Merc 450SE     16.4   8  275.8 180 3.07 4.070 17.40 0  0    3    3
## Merc 450SL     17.3   8  275.8 180 3.07 3.730 17.60 0  0    3    3
## Merc 450SLC    15.2   8  275.8 180 3.07 3.780 18.00 0  0    3    3
## Cadillac Fleetwood 10.4   8  472.0 205 2.93 5.250 17.98 0  0    3    4
## Lincoln Continental 10.4   8  460.0 215 3.00 5.424 17.82 0  0    3    4
## Chrysler Imperial 14.7   8  440.0 230 3.23 5.345 17.42 0  0    3    4
## Fiat 128       32.4   4   78.7  66 4.08 2.200 19.47 1  1    4    1
```

```
## Honda Civic      30.4   4  75.7  52 4.93 1.615 18.52  1  1   4   2
## Toyota Corolla  33.9   4  71.1  65 4.22 1.835 19.90  1  1   4   1
## Toyota Corona   21.5   4 120.1  97 3.70 2.465 20.01  1  0   3   1
## Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87  0  0   3   2
## AMC Javelin     15.2   8 304.0 150 3.15 3.435 17.30  0  0   3   2
## Camaro Z28      13.3   8 350.0 245 3.73 3.840 15.41  0  0   3   4
## Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0   3   2
## Fiat X1-9       27.3   4  79.0  66 4.08 1.935 18.90  1  1   4   1
## Porsche 914-2   26.0   4 120.3  91 4.43 2.140 16.70  0  1   5   2
## Lotus Europa    30.4   4  95.1 113 3.77 1.513 16.90  1  1   5   2
## Ford Pantera L  15.8   8 351.0 264 4.22 3.170 14.50  0  1   5   4
## Ferrari Dino    19.7   6 145.0 175 3.62 2.770 15.50  0  1   5   6
## Maserati Bora   15.0   8 301.0 335 3.54 3.570 14.60  0  1   5   8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60  1  1   4   2
```

When we print `mtcars` using `as_tibble()`, it will only print the first 10 observations. It also prints the number of rows and columns and the data type of each column.

```
as_tibble(mtcars)
```

```
## # A tibble: 32 x 11
##   mpg   cyl  disp    hp  drat    wt  qsec    vs  am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  21     6   160   110   3.9   2.62  16.5     0     1     4     4
## 2  21     6   160   110   3.9   2.88  17.0     0     1     4     4
## 3 22.8     4   108    93   3.85   2.32  18.6     1     1     4     1
## 4 21.4     6   258   110   3.08   3.22  19.4     1     0     3     1
## 5 18.7     8   360   175   3.15   3.44  17.0     0     0     3     2
## 6 18.1     6   225   105   2.76   3.46  20.2     1     0     3     1
## 7 14.3     8   360   245   3.21   3.57  15.8     0     0     3     4
## 8 24.4     4   147    62   3.69   3.19   20      1     0     4     2
## 9 22.8     4   141    95   3.92   3.15  22.9     1     0     4     2
## 10 19.2     6   168   123   3.92   3.44  18.3     1     0     4     4
## # i 22 more rows
```

To check if a data frame is tibble or not, you can use the `is_tibble()` function.

```
is_tibble(mtcars)
```

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

```
is_tibble(ggplot2::diamonds)
```

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

```
is_tibble(nycflights13::flights)
```

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

Or, you can also run `class()` to find the class of an object. Data frames that are tibble will have the classes “tbl”, “tbl_df”, and “data.frame”.

```
class(mtcars)
```

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

```
class(ggplot2::diamonds)
```

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

```
class(nycflights13::flights)
```

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

— 10.5 Exercise 2 —

Compare and contrast the following operations on a `data.frame` and equivalent tibble. What is different? Why might the default data frame behaviours cause you frustration?

```
df <- data.frame(abc = 1, xyz = "a")  
df$x
```

```
## [1] "a"
```

```
df[, "xyz"]
```

```
## [1] "a"
```

```
df[, c("abc", "xyz")]
```

```
##   abc xyz  
## 1   1   a
```

With tibble...

```
tbl <- as_tibble(df)  
tbl$x
```

```
## Warning: Unknown or uninitialised column: 'x'.
```

```
## NULL
```

```
tbl[, "xyz"]
```

```
## # A tibble: 1 x 1
##   xyz
##   <chr>
## 1 a
```

```
tbl[, c("abc", "xyz")]
```

```
## # A tibble: 1 x 2
##   abc xyz
##   <dbl> <chr>
## 1     1 a
```

What the `$` operator will do is it will match any column name that starts with the name following it. Since there is a column named `xyz`, *it will be expanded to `df$xyz`*. This is nice because it will allow you to match with fewer keystrokes, but it can also be dangerous because you can accidentally add other columns that you didn't want to add easily

With `data.frame`, with `[,]`, the type of the object that is returned differs on the number of columns. If it is one column, it won't return a `data.frame`. Instead, it will return a vector. With more than one column, it will return a `data.frame`. This is okay if you understand what you are passing in, but if not, you'd have to write code to account for those situations to avoid any bugs.

— 10.5 Exercise 4 —

Practice referring to non-syntactic names in the following data frame by:

```
annoying <- tibble(
  `1` = 1:10,
  `2` = `1` * 2 + rnorm(length(`1`))
)
```

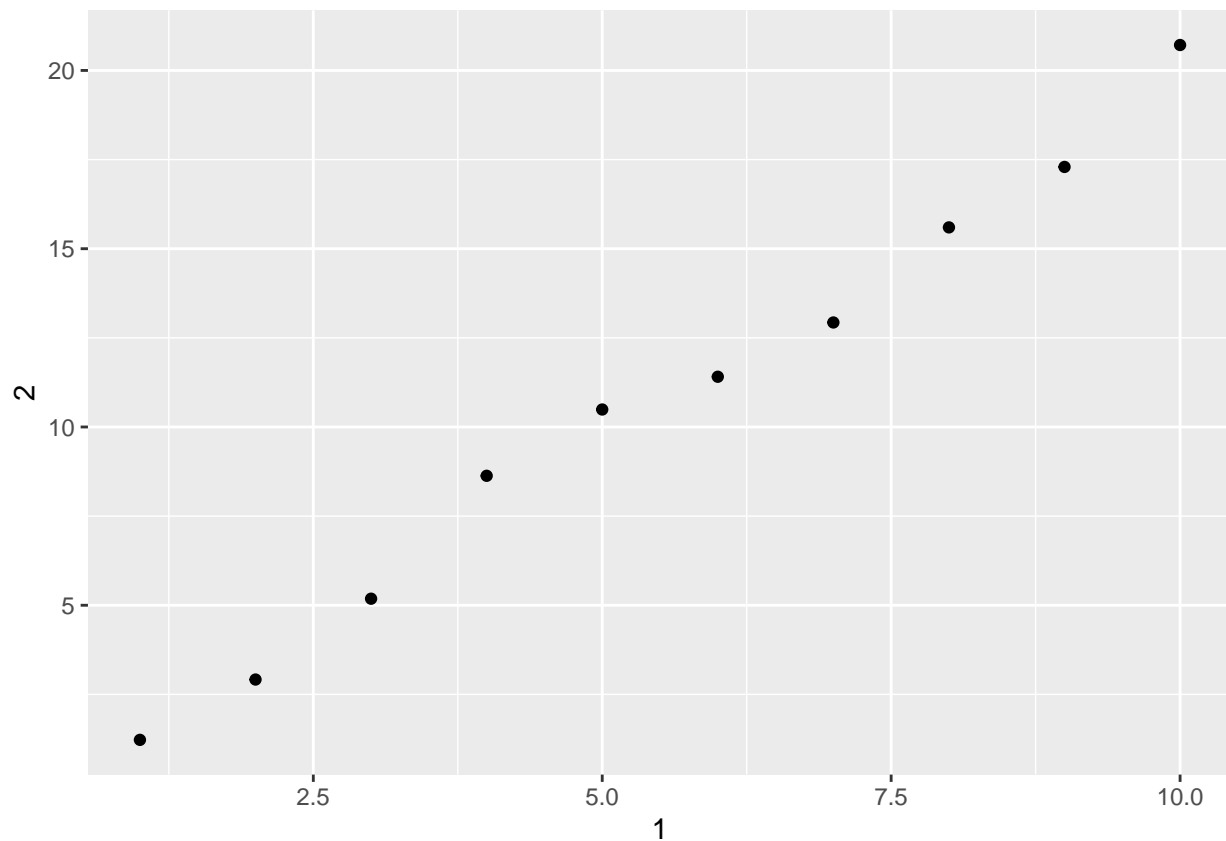
1) Extracting the variable called 1.

```
annoying$`1`
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

2) Plotting a scatterplot of 1 vs 2.

```
ggplot(annoying, aes(`1`, `2`)) +
  geom_point()
```



3) Creating a new column called 3 which is 2 divided by 1.

```
annoying <-
  annoying %>%
  mutate(`3` = `2` / `1`)
```

4) Renaming the columns to one, two and three.

```
annoying %>%
  rename(one = `1`,
         two = `2`,
         three = `3`)
```

```
## # A tibble: 10 x 3
##       one    two three
##   <int> <dbl> <dbl>
## 1     1  1.22  1.22
## 2     2  2.92  1.46
## 3     3  5.18  1.73
## 4     4  8.63  2.16
## 5     5 10.5   2.10
## 6     6 11.4   1.90
## 7     7 12.9   1.85
## 8     8 15.6   1.95
## 9     9 17.3   1.92
## 10    10 20.7   2.07
```

— 10.5 Exercise 5 —

What does `tibble::enframe()` do? When might you use it?

What `tibble::enframe()` does is it turns named vectors or lists to two-column data frames. It is different from `as_tibble()` for lists because it creates a stacked data frame rather than wide one. It all depends on your data.

```
lst <- list(female = 1, male = 2)

as_tibble(lst)
```

```
## # A tibble: 1 x 2
##   female male
##   <dbl> <dbl>
## 1     1     2
```

rather than...

```
enframe(lst) %>% unnest()
```

```
## Warning: 'cols' is now required when using 'unnest()'.
## i Please use 'cols = c(value)'.
```

```
## # A tibble: 2 x 2
##   name    value
##   <chr>   <dbl>
## 1 female     1
## 2 male      2
```

— 10.5 Exercise 6 —

What option controls how many additional column names are printed at the footer of a tibble?

The option that controls how many additional column names are printed at the footer of a tibble is `tibble::tibble.width`. This option determines the max width of a tibble before it is printed with additional column names shown at the bottom. By default, it is set to 80 characters.

```
options(tibble.width = Inf) # this is to prevent additional column names from being printed.
options(tibble.width = 120) # this is used to set a specific tibble width, in this case 120.
```