

Exercises 10.5

Tiffany Cheng

February 20, 2018

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

```
print(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

```
class(mtcars)
```

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

When a tibble is printed to the console, it will only print the first 10 lines of the data and it will mention that it is of class tibble and each column will have the type of data stored in it, such as integer, factor, etc. Here, mtcars is a data frame because such labeling does not exist.

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 # returns the column with an "x" in it

## [1] a
## Levels: a

df[, "xyz"] # returns a data frame with xyz as a factor

## [1] a
## Levels: a

df[, c("abc", "xyz")] # returns a data frame

##   abc xyz
## 1   1   a

tib <- tibble(abc = 1, xyz = "a")
tib$x # returns nothing, does not do partial matching

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

tib[, "xyz"] # returns a tibble with xyz as a character

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

tib[, c("abc", "xyz")] # returns a tibble

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

These differences can cause frustration when there are column names that have the same words in it, but you only want to select a certain column.

3. If you have the name of a variable stored in an object, e.g. `var <- "mpg"`, how can you extract the reference variable from a tibble?

```
var <- "xyz"
tib[[var]] # returns a vector

## [1] "a"

tib[var] # returns a tibble

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

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

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

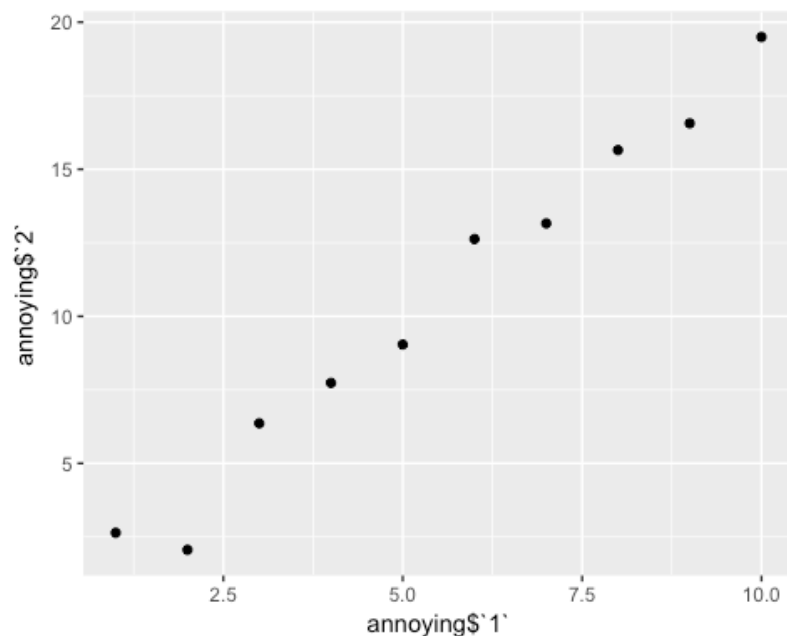
1. Extracting the variable called 1.

```
annoying$`1`

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

2. Plotting a scatterplot of 1 vs 2.

```
ggplot(data=annoying) +
  geom_point(mapping=aes(x=annoying$`1`,y=annoying$`2`))
```



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

```
annoying <-  
  annoying %>% mutate(`3` = `2` / `1`)  
  
## Warning: package 'bindrcpp' was built under R version 3.2.5  
  
annoying  
  
## # A tibble: 10 x 3  
##       `1`       `2`       `3`  
##   <int>   <dbl>   <dbl>  
## 1     1  2.639604 2.639604  
## 2     2  2.060168 1.030084  
## 3     3  6.360909 2.120303  
## 4     4  7.734426 1.933607  
## 5     5  9.040063 1.808013  
## 6     6 12.626065 2.104344  
## 7     7 13.158475 1.879782  
## 8     8 15.651216 1.956402  
## 9     9 16.560131 1.840015  
## 10    10 19.496212 1.949621
```

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

```
annoying <-  
  annoying %>% rename(one = `1`, two = `2`, three = `3`)  
  
annoying  
  
## # A tibble: 10 x 3  
##       one      two      three  
##   <int>   <dbl>   <dbl>  
## 1     1  2.639604 2.639604  
## 2     2  2.060168 1.030084  
## 3     3  6.360909 2.120303  
## 4     4  7.734426 1.933607  
## 5     5  9.040063 1.808013  
## 6     6 12.626065 2.104344  
## 7     7 13.158475 1.879782  
## 8     8 15.651216 1.956402  
## 9     9 16.560131 1.840015  
## 10    10 19.496212 1.949621
```

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

`enframe()` converts vectors to tibbles and `deframe()` does the opposite. This can be used when you have a list and would like to turn it into a dataset.

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

`tibble.max_extra_cols` controls the additional column names printed at the footer of the tibble.

```
options(tibble.max_extra_cols=2) # for example
```