

Lists and Dataframes

30 January, 2023

Write the R code to answer the following questions. You have until the beginning of next class to answer all of the questions below and commit to GitHub, both the .Rmd file and the .pdf.

Question 1 - Lists

Run this code, then answer the questions below.

```
data(cars)
mylm <- unclass(lm(dist ~ speed, data = cars))
```

1a

Is mylm a list?

Answer:

Yes! You can look several ways. `is.list()` checks if the object is a list and returns TRUE, `class()` tells you it is a list, and in the global environment it says “List of 12.”

```
is.list(mylm)
```

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

```
class(mylm)
```

```
## [1] "list"
```

1b

What is the class of the element named `model` in the `mylm` object?

Answer:

It is a `data.frame`. You can see this by using the `class` function on just the `model` object, or you can read the output of `str`, although that is a little tedious because the output is so long.

```
class(mylm[["model"]])
```

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

```
str(mylm)
```

```
## List of 12
##  $ coefficients : Named num [1:2] -17.58 3.93
##    .. attr(*, "names")= chr [1:2] "(Intercept)" "speed"
##  $ residuals    : Named num [1:50] 3.85 11.85 -5.95 12.05 2.12 ...
##    .. attr(*, "names")= chr [1:50] "1" "2" "3" "4" ...
##  $ effects      : Named num [1:50] -303.914 145.552 -8.115 9.885 0.194 ...
##    .. attr(*, "names")= chr [1:50] "(Intercept)" "speed" "" "" ...
##  $ rank         : int 2
```

```
## $ fitted.values: Named num [1:50] -1.85 -1.85 9.95 9.95 13.88 ...
## ..- attr(*, "names")= chr [1:50] "1" "2" "3" "4" ...
## $ assign      : int [1:2] 0 1
## $ qr          :List of 5
## ..$ qr       : num [1:50, 1:2] -7.071 0.141 0.141 0.141 0.141 ...
## .. ..- attr(*, "dimnames")=List of 2
## .. .. ..$ : chr [1:50] "1" "2" "3" "4" ...
## .. .. ..$ : chr [1:2] "(Intercept)" "speed"
## .. ..- attr(*, "assign")= int [1:2] 0 1
## ..$ qraux: num [1:2] 1.14 1.27
## ..$ pivot: int [1:2] 1 2
## ..$ tol   : num 1e-07
## ..$ rank  : int 2
## ..- attr(*, "class")= chr "qr"
## $ df.residual : int 48
## $ xlevels      : Named list()
## $ call         : language lm(formula = dist ~ speed, data = cars)
## $ terms        :Classes 'terms', 'formula' language dist ~ speed
## .. ..- attr(*, "variables")= language list(dist, speed)
## .. ..- attr(*, "factors")= int [1:2, 1] 0 1
## .. .. ..- attr(*, "dimnames")=List of 2
## .. .. .. ..$ : chr [1:2] "dist" "speed"
## .. .. .. ..$ : chr "speed"
## .. ..- attr(*, "term.labels")= chr "speed"
## .. ..- attr(*, "order")= int 1
## .. ..- attr(*, "intercept")= int 1
## .. ..- attr(*, "response")= int 1
## .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
## .. ..- attr(*, "predvars")= language list(dist, speed)
## .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
## .. .. ..- attr(*, "names")= chr [1:2] "dist" "speed"
## $ model        :'data.frame': 50 obs. of 2 variables:
## ..$ dist : num [1:50] 2 10 4 22 16 10 18 26 34 17 ...
## ..$ speed: num [1:50] 4 4 7 7 8 9 10 10 10 11 ...
## ..- attr(*, "terms")=Classes 'terms', 'formula' language dist ~ speed
## .. .. ..- attr(*, "variables")= language list(dist, speed)
## .. .. ..- attr(*, "factors")= int [1:2, 1] 0 1
## .. .. .. ..- attr(*, "dimnames")=List of 2
## .. .. .. .. ..$ : chr [1:2] "dist" "speed"
## .. .. .. .. ..$ : chr "speed"
## .. .. ..- attr(*, "term.labels")= chr "speed"
## .. .. ..- attr(*, "order")= int 1
## .. .. ..- attr(*, "intercept")= int 1
## .. .. ..- attr(*, "response")= int 1
## .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
## .. .. ..- attr(*, "predvars")= language list(dist, speed)
## .. .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
## .. .. .. ..- attr(*, "names")= chr [1:2] "dist" "speed"
```

1c

Change the column names of the element named `model` in the `mylm` object to Y and X.

Answer:

```
colnames(mylm[["model"]]) # what they were originally
```

```
## [1] "dist" "speed"
```

```
colnames(mylm[["model"]]) <- c("Y", "X") # changing them
```

```
colnames(mylm[["model"]]) # check it out
```

```
## [1] "Y" "X"
```

1d

Assign the element named `model` in the `mylm` object to a new object called `df`. What is the class of this object? Is it the same class or different than the object in 1b? Why?

Answer:

It is also a `data.frame`. We haven't edited the object, so the class hasn't changed. We've only assigned it to a new object outside the list.

```
df <- mylm[["model"]]
class(df)
```

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

Question 2 - Dataframes

In this exercise, you will create a new column in the data frame based on values in an existing column. Use the following code to create the data frame.

```
set.seed(123)
df <- data.frame(Y = runif(10, 0, 1),
                 X = rbinom(10, 1, .5))
df$condition <- NA
```

2a

Populate the entries of the variable `condition` with the string "Treatment" if the corresponding value of `X` is 1 and "Control" otherwise.

Answer:

I use logicals to check if `X` is 1, and if that is TRUE, I recode `condition` to "Treatment. Using another line of code, I use the same process to check if `X` is 0, and in the positions where that is TRUE in the `condition` vector, I assign "Control."

```
df$condition[df$X == 1] <- "Treatment"
df$condition[df$X == 0] <- "Control"
df
```

```
##           Y X condition
## 1 0.2875775 1 Treatment
## 2 0.7883051 0  Control
## 3 0.4089769 1 Treatment
## 4 0.8830174 1 Treatment
## 5 0.9404673 0  Control
## 6 0.0455565 1 Treatment
## 7 0.5281055 0  Control
## 8 0.8924190 0  Control
```

```
## 9 0.5514350 0 Control
## 10 0.4566147 1 Treatment
```

2b

What is the mean value of `Y` for the observations where `condition` is “Treatment”?

Answer:

Starting from the inside and working out – I first check if `condition` is equal to “Treatment”, which returns logicals. I subset to where that evaluates to TRUE and take the mean of only those 5 values. I do the same for control.

```
mean(df$Y[df$condition == "Treatment"])
```

```
## [1] 0.4163486
```

```
mean(df$Y[df$condition == "Control"])
```

```
## [1] 0.7401464
```

Question 3

In a few words, explain a few reasons you might decide to use a list vs. a dataframe (or vice versa) for a programming task. What are some different use cases for each?

Answer:

A list is much more flexible. I might use a list if writing my own function to output statistical results where the results take different sizes (i.e., a vector of coefficients will have a different length than a vector of residuals in an `lm` output). However, when working with a dataset where each row is an observation, I will most likely use a `data.frame` because the rectangular shape makes sense.