Exam format

Yufeng Huang

Associate Professor of Marketing, Simon Business School

August 15, 2022

Exam: first part

- Two parts: total is 165 minutes (including a 15-min break)
- First part (70 mins) closed book and computer
  - 60% of all points
  - basic syntax and coding logic
  - combination of logic (but not too complicated ones)
  - most questions are simple; the last few questions slightly more complex

First part: what you need to fully memorize

```
my.vector[1:6]

my.data.frame$my.variable[1:6]

my.function <- function(x, y) {
        body_of_the_function
}

for (i in 1:8) {
        do_something(i)
}

mean(c(1, 2, NA, 4), na.rm = T)

# ...
```

- Won't test your memory of detailed notations
  - so e.g. won't penalize on small writing errors (like "typo")
  - unless these are not really "typo" but an error that reflects misunderstandings

First part (both parts): what you don't need to memorize

```
# I'll hint by e.g. giving you description and notation of the function
#    but after I told you x, by and FUN you should know what aggregate() does
aggregate(x = df$trips, by = list(df$id), FUN = sum)

# I might give you a hint of the notation (but you need to know how they work)
cast(data = df, formula = x ~ y ~ z)
sapply(x, FUN = function(y) { body })

# I might recall what is grep, so you don't need to know grep vs grepl vs gsub
#    e.g. I won't give you choices between
#    '## 4' vs '## F F F T' vs '## "me"' vs '## "Hey this is "'
grep("me", unlist(strsplit("Hey this is me", sep = " ")))

# mentioned but not really emphasized in class:
optim(par, fn = function(x) { body })
plot(1:8, 2:9, col = 2, type = "l", main = "Here's a title")

# not important / too specialized
max.col(A)
which.max(A)
gender("Mary", method = 'ssa')

# ...
```

First part: example of a simple question

[Question 1]. (5 pts) Define `x` as

```
x <- "Good luck to all of you with your exams!"
```

What is the result of the following:

```
length(x)
```

(A) 9

(B) 39

(C) 41

(D) 1

First part: example of a medium-difficulty question

[Question 5]. (5 pts) Recall that function `sort` organizes elements in a vector in ascending order. Function `order` returns a vector of indices, which can be used to organize elements in ascending order. Now let's work with data frames. The `mtcars` data frame has a column for miles per gallon (`$mpg`). Which statement correctly orders rows of the data frame by `$mpg` in descending order? That is, which statement generates the following data (showing the first six rows)?

```
##                 mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## 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
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    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
```

(A) `mtcars[sort(-mtcars$mpg), ]`

(B) `mtcars[order(-mtcars$mpg), ]`

(C) `-sort(mtcars$mpg)`

(D) `-order(mtcars$mpg)`

Exam: second part

- Second part (80 mins; last year was 70 mins) open book and computer
  - 40% points
  - no internet, hand-written answers
    - [!!] no internet means that you **should have all BB material in your local computer before the exam**
    - also: please download a PDF reader before the exam and do not use a browser to read PDFs
  - four questions, all around small tasks on toy datasets
  - no data, need to type some examples yourself and see if your code works
  - increasing difficulty, but the first two questions should be relatively simple