Warmup 02: Basic Data Objects

Stat 133, Spring 2019

Introduction

The purpose of this assignment is to work with various data structures in R (e.g. vectors of different data types, factors, arrays, and lists).

Use this assignment to start developing your manipulation skills of basic data objects in R: use of bracket notation, understanding vectorization, coercion rules, recycling, etc.

General Instructions

- Write your narrative and code in an `Rmd` (R markdown) file.
- Name this file as `warmup02-first-last.Rmd`, where `first` and `last` are your first and last names (e.g. `warmup02-gaston-sanchez.Rmd`).
- Please do not use code chunk options such as: `echo = FALSE`, `eval = FALSE`, `results = 'hide'`. All chunks must be visible and evaluated.
- Submit your Rmd and html files to bCourses.

Data

The (raw) data for this assignment has to do with some players from Golden State Warriors (2018), displayed in the figure below (source: Basketball-Reference).

number	player	position	height	weight	birthdate	experience	college
30	Stephen Curry	PG	6-3	190	March 14, 1988	8	Davidson College
35	Kevin Durant	PF	6-9	240	September 29, 1988	10	University of Texas at Austin
23	Draymond Green	PF	6-7	230	March 4, 1990	5	Michigan State University
9	Andre Iguodala	SF	6-6	215	January 28, 1984	13	University of Arizona
11	Klay Thompson	SG	6-7	215	February 8, 1990	6	Washington State University
27	Zaza Pachulia	C	6-11	270	February 10, 1984	14	
34	Shaun Livingston	PG	6-7	192	September 11, 1985	12	
6	Nick Young	SG	6-7	210	June 1, 1985	10	University of Southern California
3	David West	C	6-9	250	August 29, 1980	14	Xavier University
0	Patrick McCaw	SG	6-7	185	October 25, 1995	1	University of Nevada, Las Vegas

1) Vectors

Create vectors and factors for the columns in the data table displayed above, according to the following data types. If there are missing values, codify them as `NA`.

- `number`: integer vector
- `player`: character vector
- `position`: factor
- `height`: character vector
- `weight`: real (i.e. double) vector
- `birthdate`: character vector
- `experience`: integer vector
- `college`: character vector

Use bracket notation to write R commands—displaying the output—that answer the following questions:

a) What is the name of the heaviest player?

b) What is the college of the player that has a height of `6-6`?

c) What is the position of the player with more years of experience?

d) What is the number of the lightest player?

e) Obtain frequencies (counts) of the position values.

f) How many players have a weight larger than the average (i.e. mean) weight?

g) How many players have between 9 and 12 years of experience (inclusive)?

h) What is the mean years of experience of *Shooting Guard* (`SG`) players?

i) What is the median weight of those players with a position different of *Center* (`C`)?

j) What is the first quartile (i.e. bottom 25th percentile) of years of experience among *Power Forwards* (`PF`) and *Shooting Guards* (`SG`)

2) List for GSW

Use the vectors created in the previous section to create the following list `gsw`:

```
gsw <- list(
  player = player,
  number = number,
  position = position,
  weight = weight,
  experience = experience
)
```

Use the list `gsw` to write R commands—displaying the output—that answer the following questions (use only the list `gsw`, NOT the individual vectors):

a) What is the number of the heaviest player?

b) What is the position of the player with less experience?

c) How many players have less than 8 or more than 11 years of experience?

d) What is the third quartile (i.e. bottom 75th percentile) of years of experience among *Power Forwards* (`PF`) and *Shooting Guards* (`SG`)

e) What is the name of the player whose weight is furthest from the average weight (of all players)?

3) More lists

Consider the following list:

```
hp <- list(
  first = 'Harry',
  last = 'Potter',
  courses = c('Potions', 'Enchantments', 'Spells'),
  sport = 'quidditch',
  age = 18L,
  gpa = 3.9
)
```

Write R commands—displaying the output—to answer the following questions:

a) What is the class of `hp`?

b) How many elements are in `hp`?

c) What is the length of `courses`?

d) What is the data type of the element `age`?

e) What is the data type of the element `gpa`?

f) If you combine `age` and `gpa` in a new vector, what is the data type of this vector?

4) Technical Questions

a) Explain why the following command returns `2`?

```
1 + TRUE
```

b) Explain why the following command returns `FALSE`?

```
"-2" > 0
```

c) Explain why the following command returns `TRUE`?

```
(10 <= 5) >= 0
```

d) Explain why the following commands A and B are not equivalent?

```r
# command A
1 + !TRUE

# command B
!TRUE + 1
```

5) Subsetting

Consider the following vector `lord`:

```r
lord <- c('v', 'o', 'l', 'd', 'e', 'm', 'o', 'r', 't')
```

Run the following commands and explain what's happening in each of them (in terms of subsetting, coercion, recycling, vectorization, etc):

a) `lord[TRUE]`

b) `lord[length(lord) + 1]`

c) `lord[seq(from = length(lord), to = 1, by = -2)]`

d) `lord[lord == "o"]`

e) `lord[lord != "e" & lord != "o"]`

f) `lord[lord %in% c('a', 'e', 'i', 'o', 'u')]`

g) `toupper(lord[!(lord %in% c('a', 'e', 'i', 'o', 'u'))])`

h) `paste(lord, collapse = '')`

i) `lord[is.na(lord)]`

j) `sum(!is.na(lord))`

6) 2D Arrays (i.e. Matrices)

Use the vector `lord` to create a matrix `vol` with 3 rows and 3 columns, like the one displayed below.

```
     [,1] [,2] [,3]
[1,] "v"  "d"  "o"
[2,] "o"  "e"  "r"
[3,] "l"  "m"  "t"
```

Use bracket notation and the matrix `vol` to write R commands—displaying the output—to:

a) obtain the following output

```
[1] "v" "d" "o"
```

b) obtain the following output

```
     [,1] [,2]
[1,] "d"  "v"
[2,] "e"  "o"
```

c) obtain the following output

```
     [,1] [,2] [,3]
[1,] "l"  "m"  "t"
[2,] "o"  "e"  "r"
[3,] "v"  "d"  "o"
```

d) obtain the following output

```
     [,1] [,2] [,3]
[1,] "v"  "d"  "d"
[2,] "o"  "e"  "e"
[3,] "l"  "m"  "m"
```

e) obtain the following output

```
     [,1] [,2] [,3]
[1,] "t"  "m"  "l"
[2,] "r"  "e"  "o"
[3,] "o"  "d"  "v"
```

f) obtain the following output

```
     [,1] [,2] [,3] [,4]
[1,] "t"  "m"  "m"  "t"
[2,] "r"  "e"  "e"  "r"
[3,] "o"  "d"  "d"  "o"
[4,] "o"  "d"  "d"  "o"
[5,] "r"  "e"  "e"  "r"
[6,] "t"  "m"  "m"  "t"
```

g) obtain the following output

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] "l"  "m"  "t"  "t"  "m"  "l"
[2,] "o"  "e"  "r"  "r"  "e"  "o"
[3,] "v"  "d"  "o"  "o"  "d"  "v"
[4,] "v"  "d"  "o"  "o"  "d"  "v"
[5,] "o"  "e"  "r"  "r"  "e"  "o"
[6,] "l"  "m"  "t"  "t"  "m"  "l"
```