M13-Quiz

learningSpoonsR@gmail.com

I. Variable & Function

Problem 1.

```
a <- "Hello"
a
```

```
## [1] "Hello"
```

ans:

Problem 2.

```
a <- "Hello"
b <- "World"
paste(a,b)
```

```
## [1] "Hello World"
```

ans:

Problem 3.

```
paste0(a,b)
```

```
## [1] "HelloWorld"
```

ans:

Problem 4.

```
paste(a,b,sep="-")
```

```
## [1] "Hello-World"
```

ans:

II. Data Type

String

Problem 1.

```
font <- "The quick brown fox"
substr(font, 4, 7)
```

```
## [1] " qui"
```

ans:

Problem 2.

```
nchar(font)
```

```
## [1] 19
```

ans:

- 아래 코드를 입력해보세요. 에러를 읽어보세요.

```
library(stringr)
```

- 위의 명령을 실행시키려면 해당 컴퓨터에 아래와 같이 `stringr`이라는 `library(package)`를 `install` 해주어야 합니다. 아래 명령을 실행하세요. (따옴표를 꼭 입력해야 합니다.)

```
install.packages("stringr")
```

- 다시 아래코드를 입력하여 에러가 발생하지 않는 것을 확인하세요.

```
library(stringr)
```

Problem 3.

```
# In `font`, replace "brown" by "white"
str_replace(font, "brown", "white")
```

```
## [1] "The quick white fox"
```

ans:

Problem 4. 위의 명령은 다음과 같이 **argument**(인수)를 명시하여 사용할 수도 있습니다. 코드가 길어지는 단점과 가독성이 높아지는 장점이 있습니다.

```
str_replace(string = font, pattern = "brown", replacement = "white")
```

```
## [1] "The quick white fox"
```

ans:

Problem 5.

```
str_detect(string = font, pattern = "fox")
```

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

```
str_detect(string = font, pattern = "rabbit")
```

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

- Github → **M6X** → **04. strings.pdf**에 더 많은 string관련 함수가 소개되어 있습니다.

Date

Problem 6.

```
aprilFool <- "2018-04-01"
class(aprilFool)
```

```
## [1] "character"
```

ans:

Problem 7.

```
as.Date(aprilFool)
```

```
## [1] "2018-04-01"
```

```
class(aprilFool)
```

```
## [1] "character"
```

ans:

Problem 8.

```
aprilFool <- as.Date(aprilFool)
class(aprilFool)
```

```
## [1] "Date"
```

ans:

- 아래 명령이 에러를 발생시키지 않게 하려면 어떤 명령을 먼저 실행해야 할까요?

```
library(lubridate)
```

- 정답은 위의 `stringr`에서 처럼 `install.packages("lubridate")` 입니다.

Problem 9.

```
myDate <- as.Date("2018-03-26")
floor_date(myDate, "month")
```

```
## [1] "2018-03-01"
```

ans:

Problem 10.

```
floor_date(myDate, "month") - 1
```

```
## [1] "2018-02-28"
```

Problem 11. 위의 두 문제에서 `myDate`의 '이번달 1일'과 '저번달 말일'을 찾아냈습니다. '저번달 1일'과 '전년 말일'을 출력하려면 어떻게 해야할 까요?

```
## floor_date(floor_date(myDate, "month")-1, "month")
## floor_date(myDate, "year")-1
```

- Github → `M6X` → `05. lubridate.pdf`에 더 많은 `Date`관련 함수가 소개되어 있습니다.

Problem 12. Cheatsheet을 참고해서 '전년 동분기 말일'을 찾아보세요.

```
## ceiling_date(myDate - years(1), "quarter")-1
```

III. Data Structure

`vector` (강의노트와 `base` cheatsheet을 이용해서 풀어보세요)

```
x <- c(2, 5, 6, 7, 8, 5, 2, 1, 6)
```

Problem 1. `x`에는 몇 개의 원소가 포함되어 있습니까?
```
## [1] "length(x)"
```
ans:

Problem 2. `x`에서 중복값을 제거하여 원소를 나열하세요.
```
## [1] "unique(x)"
```
ans:

Problem 3. `x`에서 중복값을 제거한 원소의 갯수는 몇 개 입니까?
```
## [1] "length(unique(x))"
```
ans:

Problem 4. `x`의 각 원소가 3보다 큰지 아닌지 보여주세요.
```
## [1] "x>3"
```
ans:

Problem 5. `x`에서 3보다 큰 원소만 보여주세요.
```
## [1] "x[x>3]"
```
ans:

Problem 6. `x`에서 3보다 큰 원소는 각각 몇 번째에 존재합니까?
```
## [1] "which(x>3)"
```
ans:

Problem 7. `x`에서 3보다 큰 원소의 갯수는 몇 개 입니까?
```
## [1] "length(which(x>3)) or sum(x>3)"
```
ans:

Problem 8. `x`의 두 번째 원소는 전체 원소 중에서 몇 번째 입니까?
```
## [1] "sum(x>x[2]) + 1"
```
ans:

data.frame

Problem 1

- 아래의 코드를 입력해보세요.

```
players <- data.frame(
  name = c("HMSon", "JSPark"),
  id = c("920708-1234567", "810225-1357913"),
  stringsAsFactors = FALSE
)
players
```

```
##     name              id
## 1  HMSon 920708-1234567
## 2 JSPark 810225-1357913
```

- **players$id**에서 맨 앞의 두 자를 이용해서 출생년도를 만들었습니다.

```
players$year <-
  paste0("19",
         substr(players$id, 1, 2))
players
```

```
##     name              id year
## 1  HMSon 920708-1234567 1992
## 2 JSPark 810225-1357913 1981
```

- 비슷한 방법으로 **player$month**와 **player$day**도 만들어보세요.
- 아래와 같이 만들어 지나요?

```
players
```

```
##     name              id year month day
## 1  HMSon 920708-1234567 1992    07  08
## 2 JSPark 810225-1357913 1981    02  25
## players$month <- substr(players$id,3,4)
## players$day <- substr(players$id,5,6)
```

Problem 2

- 아래 `ifelse()` 함수는 처음 나왔지만 이해되시나요?
- R의 많은 문법이 상식적이고 엑셀과도 비슷한 점이 많습니다.

```
players$sex <- ifelse(substr(players$id, 8, 8)=="1", "Male", "Female")
players$region <- substr(players$id, 9, 12)
players
```

```
##     name              id year month day  sex region
## 1  HMSon 920708-1234567 1992    07  08 Male   2345
## 2 JSPark 810225-1357913 1981    02  25 Male   3579
```

- `paste()`함수와 `year`, `month`, `date`를 이용해서 `birth_date`를 `"YYYY-MM-DD"` 형식으로 만들어보세요.

```
players
```

```
##     name              id year month day  sex region birth_date
## 1  HMSon 920708-1234567 1992    07  08 Male   2345 1992-07-08
## 2 JSPark 810225-1357913 1981    02  25 Male   3579 1981-02-25
## players$birth_date <- paste(players$year, players$month, players$day, sep = "-")
```

Problem 3

- 데이터 셋이 거의 다 정리되었습니다. 관심있는 변수만 모으고 각 변수의 type을 확인합니다.

```
players <- players[,c(1, 8, 6, 7)]
players
```

```
##     name birth_date  sex region
## 1  HMSon 1992-07-08 Male   2345
## 2 JSPark 1981-02-25 Male   3579
```

```
sapply(players, class)
```

```
##        name  birth_date         sex      region
## "character" "character" "character" "character"
```

- `birth_date`는 `Date`형으로, `sex`는 `factor`형으로 바꾸어 보세요.

```
## players$birth_date <- as.Date(players$birth_date)
## players$sex <- as.factor(players$sex)
```

- 성공적으로 수행한다면 아래와 같은 결과를 확인할 수 있습니다.

```
sapply(players, class)
```

```
##        name  birth_date         sex      region
## "character"      "Date"    "factor" "character"
```

IV. Function

- 2차원 평면에서 점 `(x,y)`와 원점과의 거리는 $\sqrt{x^2 + y^2}$ 입니다. 이를 함수로 표현하면 다음과 같습니다.

```
dist <- function(x, y) {
  return(sqrt(x^2 + y^2))
}
```

- 이를 `M12`에서처럼 도식화하여 input과 output을 명시하여 그림을 그려보세요.
- 위의 코드를 입력하고 `dist(3,4)`를 입력해보세요. `5`라는 값이 나와야 합니다.