

# 중간고사

\_\_\_\_\_학과 학번\_\_\_\_\_이름\_\_\_\_\_

2017-04-18

## I. Dice

`die <- 1:6` 를 실행하였다. 다음 물음에 답하십시오.

1. 다음 코드 중 에러 메시지가 나올 것으로 예상되는 것은?

가. `die + 1`

나. `die + 1:2`

다. `die + 1:3`

라. `die + 1:4`

2. 다음 중 그 결과가 다른 것을 하나 고르시오.

가. `die * die`

나. `die ^ 2`

다. `die ** 2`

라. `die %**% die`

3. `die / die` 의 계산 결과는 콘솔에 어떻게 나타나는가?

가. 1

나. `[1] 1`

다. `1 1 1 1 1 1`

라. `[1] 1 1 1 1 1 1`

4. `die` 의 각 원소를 5로 나눈 나머지를 나타내는 코드는 어느 것인가?

가. `die %% 5`

나. `die %/% 5`

다. `die %**% 5`

라. `die %o% 5`

5. 사칙 연산자 중 \* 에 대한 도움말을 구하고자 할 때 옳지 않은 것은?

가. `help(*)`

나. `help("*")`

다. `?`*``

라. `? "*"`

6. 다음 중 나머지 셋과 다른 연산을 수행하고 있는 것은?

- 가. `die %% 2`
- 나. `%%"(die, 2)`
- 다. ``%%` (die, 2)`
- 라. ``%%` (2, die)`

7. 공평한 주사위를 두 번 굴릴 때 나오는 결과를 표현한 코드로 적절한 것은?

- 가. `sample(2, die)`
- 나. `sample(die, 2, replace = TRUE)`
- 다. `sample(die, 2, replace = FALSE)`
- 라. `sample(2, die, replace = TRUE)`

8. `roll2` 를 다음과 같이 정의한 함수라 할 때,

```
roll2 <- function(bones) {  
  dice <- sample(bones, size = 2, replace = TRUE)  
  sum(dice)  
}
```

어떤 경우에 에러가 출력되는가?

- 가. `roll2(bones = 1:4)`
- 나. `roll2(bones = 1:5)`
- 다. `roll2(1:20)`
- 라. `roll2()`

9. 위에서 정의한 `roll2` 를 나중에 다시 불러들여 사용하기 위하여 저장할 때 적절한 코드는?

- 가. `dump(list = "roll2", file = "./roll2.R")`
- 나. `save(list = "roll2", file = "./roll2.R")`
- 다. `save.image(file = "./roll2.R")`
- 라. `write(list = "roll2", file = "./roll2.R")`

10. 위와 같이 저장한 `roll2` 를 불러들이는 코드는 무엇인가? 작업디렉토리와 저장한 티렉토리가 같다고 가정하자.

- 가. `read("./roll2.R")`
- 나. `source("./roll2.R")`
- 다. `load("./roll2.R")`
- 라. `get("./roll2.R")`

## II. Simulation

11. 공평한 주사위 두 개를 던져 합을 구하는 게임을 20번 반복하는 코드는 다음 중 어느 것인가? 위에서 정의한 `roll2` 를 이용하시오. 단, `roll2` 는 위에 정의한 대로 임.

- 가. `replicate(20, roll2())`
- 나. `replicate(20, roll2(bones = 1:6))`
- 다. `replicate(roll2(), 20)`
- 라. `replicate(roll2(bones = 1:6), 20)`

12. 위의 실험을 1만번 반복하고, 그 결과를 `rolls` 에 저장한 후 `ggplot` 패키지를 이용하여 간단한 히스토그램으로 나타내고자 한다. `ggplot` 패키지는 설치되어 있고 `library` 에 등록되어 있을 때 자료 특성에 비추어 가장 적절한 코드는?

- 가. `plot(rolls)`
- 나. `plot(rolls, binwidth = 1)`
- 다. `qplot(rolls)`
- 라. `qplot(rolls, binwidth = 1)`

13. `roll` 함수를 편집하여 확률 조건을 추가한 `roll.w` 를 만들려고 할 때 필요한 코드는? (단, 여기서 `;` 는 그 다음 코드를 이어서 수행하라는 것이다.)

- 가. `roll.w <- roll ; fix(roll.w)`
- 나. `roll.w <- roll ; vi(roll.w)`
- 다. `roll <- roll.w ; fix(roll.w)`
- 라. `roll <- roll.w ; vi(roll.w)`

## III. R Objects

14. `is.vector(5)` 의 답은 무엇인가?

- 가. `TRUE`
- 나. `FALSE`
- 다. `Yes`
- 라. `No`

15. 위에서 저장해 놓은 `die` 에 대해서 `typeof(die)` 의 값은 무엇인가?

- 가. `"logical"`
- 나. `"integer"`
- 다. `"double"`
- 라. `"complex"`

16. `attributes(die)` 를 수행했더니 `$names` 항목이 발견되었다. 이 속성을 없애고자 할 때 적당한 코드는?

가. `names(die) <- ""`

나. `names(die) <- NA`

다. `names(die) <- NULL`

라. `names(die) <- NaN`

17. `dim(die) <- c(2, 3)` 를 수행하였을 때 `class(die)` 는 무엇인가?

가. `[1] "integer"`

나. `[1] "vector"`

다. `[1] "matrix"`

라. `[1] "array"`

18. `today <- Sys.Date()` ; `class(today)` 의 결과는?

가. `[1] "integer"`

나. `[1] "Date"`

다. `[1] "matrix"`

라. `[1] "array"`

19. `gender <- factor(c("male", "female", "female", "male"))` 를 수행하고 `unclass(gender)` 를 수행하면 그 결과는?

가. `[1] 2 1 1 2`

나. `[1] "integer"`

다.

```
[1] male    female female male
Levels: female male
```

라.

```
$levels
[1] "female" "male"

$class
[1] "factor"
```

## IV. list and data frame

- list2 <- list(number = 100:130, char = "R", logical = list(TRUE, FALSE)) 에서

20. list2[[2]] 의 결과는 무엇인가?

가.

```
$char  
[1] "R"
```

나. [1] "R"

다.

```
$R  
[1] "char"
```

라. [1] "char"

21. list2[2] 의 결과는 무엇인가?

가.

```
$char  
[1] "R"
```

나. [1] "R"

다.

```
$R  
[1] "char"
```

라. [1] "char"

22. 다음 중 그 결과가 다른 것 하나를 고르시오.

가. list2\$logical

나. list2[[3]]

다. list2[3]

라. list2[3][[1]]

23. logical 중 첫번째 TRUE 를 불러내는 코드는 무엇인가? (즉, [1] TRUE 가 나오는 코드는?)

가. list2[3][[1]][[1]]

나. list2[3][[1]][1]

다. list2[3][1][1]

라. list2[3][1][[1]]

- df 를 다음과 같이 정의할 때

```
df <- data.frame(face = c("ace", "two", "six"),
                 suit = rep("clubs", 3),
                 value = c(1, 2, 6))
```

24. str(df) 의 결과를 올바르게 나타낸 것은?

가.

```
'data.frame':  3 obs. of  3 variables:
 $ face : chr  "ace" "two" "six"
 $ suit : chr  "clubs" "clubs" "clubs"
 $ value: num  1 2 6
```

나.

```
'data.frame':  3 obs. of  3 variables:
 $ face : Factor w/ 3 levels "ace","two","six": 1 2 3
 $ suit : Factor w/ 1 level "clubs": 1 1 1
 $ value: num  1 2 6
```

다.

```
'data.frame':  3 obs. of  3 variables:
 $ face : Factor w/ 3 levels "ace","six","two": 1 3 2
 $ suit : Factor w/ 1 level "clubs": 1 1 1
 $ value: num  1 2 6
```

라.

```
'data.frame':  3 obs. of  3 variables:
 $ face : chr  "ace","two","six": 1 2 3
 $ suit : chr  "clubs": 1 1 1
 $ value: num  1 2 6
```

25. df 의 정의에 `stringsAsFactors = FALSE` 를 추가하면 `str(df)` 는 어떻게 나오는가?

가.

```
'data.frame':  3 obs. of  3 variables:
 $ face : chr  "ace" "two" "six"
 $ suit : chr  "clubs" "clubs" "clubs"
 $ value: num  1 2 6
```

나.

```
'data.frame':  3 obs. of  3 variables:
 $ face : Factor w/ 3 levels "ace","two","six": 1 2 3
 $ suit : Factor w/ 1 level "clubs": 1 1 1
 $ value: num  1 2 6
```

다.

```
'data.frame':  3 obs. of  3 variables:
 $ face : Factor w/ 3 levels "ace","six","two": 1 3 2
 $ suit : Factor w/ 1 level "clubs": 1 1 1
 $ value: num  1 2 6
```

라.

```
'data.frame':  3 obs. of  3 variables:
 $ face : chr  "ace","two","six": 1 2 3
 $ suit : chr  "clubs": 1 1 1
 $ value: num  1 2 6
```

- 다음 그림과 같은 구조를 갖는 cards.csv 의 URL이  
[https://raw.githubusercontent.com/coop711/r\\_programming/master/data/cards.csv](https://raw.githubusercontent.com/coop711/r_programming/master/data/cards.csv) 라고 하자.

```
"face", "suit", "value"
"king", "spades", 13
"queen", "spades", 12
"jack", "spades", 11
"ten", "spades", 10
"nine", "spades", 9
```

26. 이를 읽어서 deck 에 저장하는 코드로 적절한 것은? 단, string 은 character 로 읽어들이는다.

가.

```
url <- "https://raw.githubusercontent.com/coop711/r_programming/master/data/cards.csv"
deck <- read.csv(url)
```

나.

```
url <- https://raw.githubusercontent.com/coop711/r_programming/master/data/cards.csv
deck <- read.csv(url)
```

다.

```
url <- "https://raw.githubusercontent.com/coop711/r_programming/master/data/cards.csv"
deck <- read.csv(url, stringsAsFactors = FALSE)
```

라.

```
url <- https://raw.githubusercontent.com/coop711/r_programming/master/data/cards.csv
deck <- read.csv(url, stringsAsFactors = FALSE)
```

27. 위와 같이 읽어들이는 deck 을 작업 디렉토리와 같은 위치에 있는 data 폴더에 cards.csv 라는 이름으로 저장하고자 한다. 단, 첫 열에 나오는 일련번호를 없애고자 할 때 적절한 코드는?

가. write.csv(deck, file = "../data/cards.csv")

나. write.csv(deck, file = "../data/cards.csv", row.names = FALSE)

다. save(deck, file = "../data/cards.csv")

라. save(deck, file = "../data/cards.csv", row.names = FALSE)

28. 현재 작업 디렉토리에 있는 모든 오브젝트를 r\_objects.RData 라는 파일로 저장하고자 한다. 적절하지 않은 것은?

가. save(list = ls(), file = "./r\_objects.RData")

나. save.image(list = ls(), file = "./r\_objects.RData")

다. save.image(file = "./r\_objects.RData")

라. save.image("./r\_objects.RData")



29. 위에서 저장한 `r_objects.RData` 를 불러오는 코드로 적절한 것은?

- 가. `load("./r_objects.RData")`
- 나. `read("./r_objects.RData")`
- 다. `get("./r_objects.RData")`
- 라. `dget("./r_objects.RData")`

## V. Notation

30. 앞에서 정의한 `deck` 에서 맨 처음 카드를 뽑아 `face`, `suit`, `value` 를 다 보고자 한다. 이 때 필요한 코드는?

- 가. `deck[1, ]`
- 나. `deck[, 1]`
- 다. `deck[1, 1]`
- 라. `deck[1, 1, drop = FALSE]`

31. 다음 중 그 효과가 나머지 셋과 같지 않은 것은?

- 가. `sample(1:52, size = 52, replace = TRUE)`
- 나. `sample(1:52, size = 52)`
- 다. `sample(1:52)`
- 라. `sample(52)`

## VI. Modifying Values

32. `vec <- 1:6` 에 7번째 원소를 추가하려다가 실수로 `vec[8] <- 7` 이라고 입력하여 7번째 원소만 없애고자 할 때 적합한 코드는 ?

- 가. `vec[7] <- ""`
- 나. `vec[7] <- NULL`
- 다. `vec[7] <- NaN`
- 라. `vec <- vec[-7]`

33. 위의 `deck` 에서 매 13번째 카드의 `value` 를 찾아보는 코드로 적합치 않은 것은?

- 가. `deck2$value[seq(13, 52, by = 13), ]`
- 나. `deck2[seq(13, 52, by = 13), "value"]`
- 다. `deck2[seq(13, 52, by = 13), 3]`
- 라. `deck2[seq(13, 52, by = 13), c(FALSE, FALSE, TRUE)]`

34. `c(1, 2, 3, 4) %in% c(3, 4, 5)` 의 결과는 ?

- 가. `[1] TRUE TRUE FALSE`
- 나. `[1] FALSE FALSE TRUE TRUE`
- 다. `[1] TRUE TRUE FALSE FALSE`
- 라. `[1] 3 4`

35. `a <- c(1, 2, 3)`, `b <- c(1, 2, 3)`, `c <- c(1, 2, 4)` 라 할 때 `a == b & b == c` 의 결과는 무엇인가?

가. [1] TRUE TRUE TRUE

나. [1] TRUE TRUE FALSE

다. [1] TRUE FALSE TRUE

라. [1] FALSE TRUE TRUE

36. 잘 섞인 카드 deck3 에서 ace 가 몇 번째 자리에 있는지 알아보는 코드로 적절한 것은 다음 중 어느 것인가?

가. deck3\$face == "ace"

나. which(deck3\$face == "ace")

다. deck3\$face = "ace"

라. which(deck3\$face = "ace")

37. 위의 deck3 에서 ace 가 들어 있는 카드를 모두 뽑아 보는 코드로 적합한 것은?

가. deck3[deck3\$face == "ace", ]

나. deck3[deck3\$face = "ace", ]

다. deck3[deck3\$face == "ace"]

라. deck3[deck3\$face = "ace"]

38. 위의 deck3 에서 face 가 king, queen, jack 인 것들의 위치를 찾기 위한 코드로 적절한 것은?

가. deck3\$face == c("king", "queen", "jack")

나. deck3\$face %in% c("king", "queen", "jack")

다. deck3\$face = c("king", "queen", "jack")

라. deck3\$face && c("king", "queen", "jack")

39. 위의 위치를 facecard 라는 R object로 저장하였다. 이 카드들의 value 를 모두 10으로 바꾸는 코드로 적절치 않은 것은?

가. deck3[facecard, "value"] <- 10

나. deck3\$value[facecard] <- 10

다. deck3\$value[facecard, ] <- 10

라. deck3[facecard, 3] <- 10

40. 다음 중 그 연산의 결과가 다른 것은?

가. any(TRUE, TRUE, NA)

나. all(TRUE, TRUE, NA)

다. any(FALSE, TRUE, NA)

라. all(TRUE, TRUE, TRUE)