SDS Review Ch10

2017100057 / 이영노

October 15, 2022

Chapter10: Probabilities

- example1: Monte Carlo simulations for categorical data

```
B <- 10000
beads <- rep(c("red","blue"),times=c(2,3)) # blue 60% red 40%
beads
```

```
## [1] "red"  "red"  "blue" "blue" "blue"
```

```
set.seed(123)
events <- replicate(B,sample(beads,1))
tab=table(events)
tab
```

```
## events
## blue  red
## 5886 4114
```

```
prop.table(tab)
```

```
## events
##   blue    red
## 0.5886 0.4114
```

```
events <- sample(beads,B,replace=TRUE) # 비복원추출로 sample 한개 뽑은거임
events=table(events)
prop.table(events)
```

```
## events
##   blue    red
## 0.6078 0.3922
```

- example2: Combinations and Permuations

```
library("gtools")
combinations(4,2)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    3
## [3,]    1    4
```

```
## [4,]    2    3
## [5,]    2    4
## [6,]    3    4
```

```
class(combinations(4,2))
```

```
## [1] "matrix" "array"
```

```
permutations(4,2)
```

```
##       [,1] [,2]
##  [1,]    1    2
##  [2,]    1    3
##  [3,]    1    4
##  [4,]    2    1
##  [5,]    2    3
##  [6,]    2    4
##  [7,]    3    1
##  [8,]    3    2
##  [9,]    3    4
## [10,]    4    1
## [11,]    4    2
## [12,]    4    3
```

Example1: Monty Hall Problem

```
B <- 10000
monty_hall <- function(strategy){
  doors <- as.character(1:3)
  prize <- sample(c("car","goat","goat")) # 정답(저렇게 샘플을 뽑음, 순서 바뀔수도)
  prize_door <- doors[prize =="car"] # TRUE/FALSE indexing.
  my_pick <- sample(doors,1) # 선택지(doors 하나를 뽑음)
  show <- sample(doors[!doors %in% c(my_pick,prize_door)],1) # mypick, prize제외 보여줌.
  stick <- my_pick
  # 인자값 strategy를 두개로 나눔: "stick", "switch"
  # 결과값을 stick, switch, choice에 전달
  switch <- doors[!doors%in% c(my_pick,show)]
  choice <- ifelse(strategy == "stick", stick ,switch)
  choice == prize_door
}


stick = replicate(B,monty_hall("stick"))
mean(stick)
```

```
## [1] 0.3306
```

```
switch = replicate(B, monty_hall("switch"))
mean(switch)
```

```
## [1] 0.6661
```

Example2: Birthday Problem

```
n <- 50 # 학생 수
bdays <- sample(1:365,n,replace=TRUE) # 365개 다른 sample중 n개를, 복원추출로 뽑음
# 할당을 해줌으로써 seed를 고정시킴

duplicated(c(1,2,3,1,4,3,5))
```

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

```
any(duplicated(bdays))
```

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

```
# duplicate 를 통해 몇개가 겹쳐있는지 쉽게 알수있음

B <- 10000
same_birthday <- function(n){
  bdays <- sample(1:365, n, replace=TRUE)
  any(duplicated(bdays))
}
# (1) function 정의
results <- replicate(B,same_birthday(50)) # 50명중 하나라도 겹치는 생일이 있음?
mean(results)
```

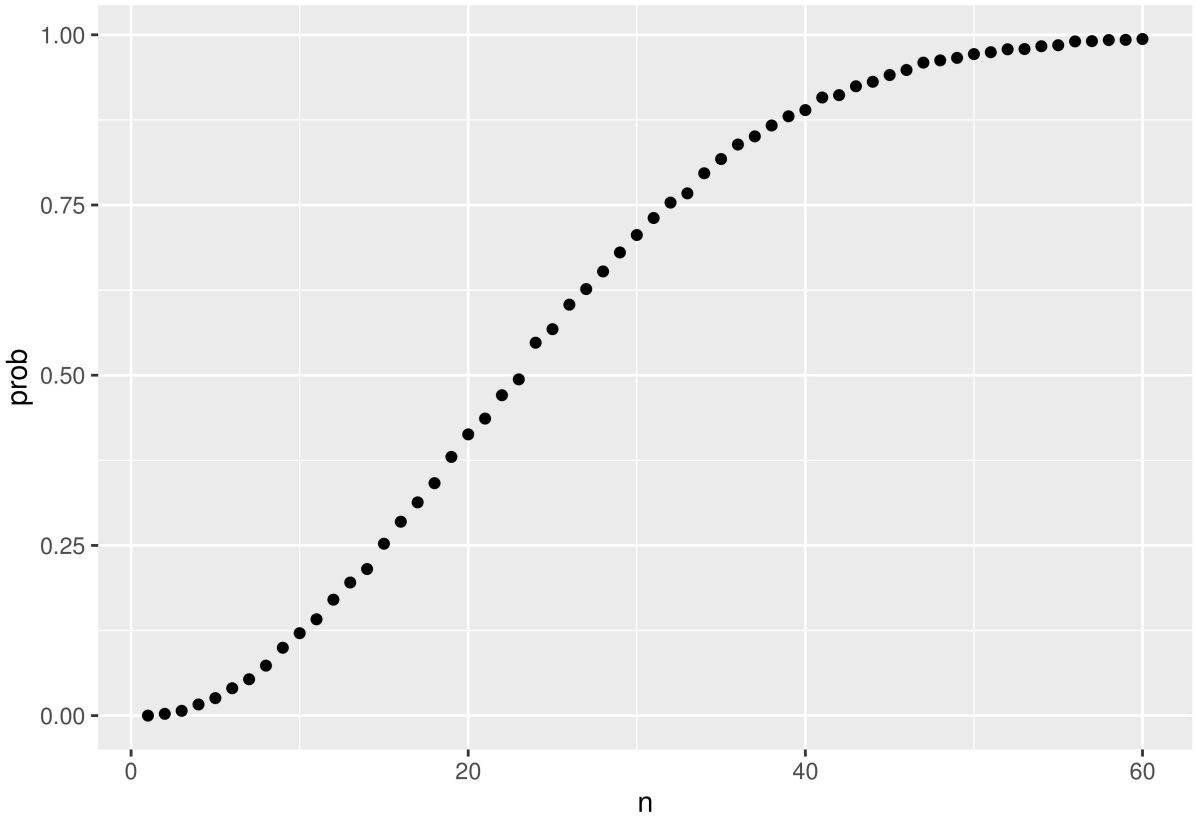
```
## [1] 0.9719
```

```
# (2) replicate with hyperparameter : n
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.7      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
compute_prob <- function(n,B=10000){
  results=replicate(B,same_birthday(n))
  mean(results)
}
# (3) compute uncertainty with change in hyperparameter: n
n=seq(1,60)
prob<-sapply(n, compute_prob)
qplot(n,prob) #library(tidyverse) 있어야 qplot돌아감
```



Example3: CDF

- CDF

```
library(dslabs)
data(heights)
x <- heights%>% filter(sex=="Male") %>% pull(height) # male height
F <- function(a) mean(x<=a) # TRUE/FALSE indexing이 되서, mean을 하면 true의 비율 나타냄.

1-F(70.5) # 임의로 뽑았을때 70.5보다 키가 클확률
```

```
## [1] 0.3633005
```

- Theoretical continuous distribution

```
## Normal Approximation of heights
m = mean(x)
s = sd(x)
1 - pnorm(70.5,m,s)
```

```
## [1] 0.371369
```

```
## height follows Normal Distribution (전제)
```

Example4: Monte Carlo simulations for continuous variables

```
n = length(x)
m = mean(x)
s = sd(x)
simulated_heights = rnorm(n,m,s) # random number generation from EXACT Normal D
```

- 응용

```
B=10000
tallest = replicate(B,{
  simulated_data = rnorm(800,m,s)
  max(simulated_data)
})
mean(tallest >= 7*12)
```

```
## [1] 0.0197
```

```
data.frame(tallest=tallest)%>%ggplot(aes(tallest))+
  geom_histogram(color="black",binwidth=1)
```