C1_DTMC Python

Jaemin Park

2021-01-03

차 례

Exercises

p.27 DTMC Simulator

R code

```
soda_simul <- function(this_state) {
u <- runif(1)
if (this_state == "c") {
if (u <= 0.7) {
next_state <- "c"
}
else {
next_state <- "p"
}
} else {
if (u <= 0.5) {
next_state <- "c"
}
else {
next_state <- "p"
}
}
return(next_state)
}
library(stringr) # for str_sub() and str_count()
for (i in 1:5) {
path <- "c" # coke today (day-0)
for (n in 1:9) {
this_state <- str_sub(path,-1,-1) # last element
next_state <- soda_simul(this_state)
path <- paste0(path, next_state)
}
print(path)
}
```

Python code

```
def soda_simul(this_state):
```

```
    u = np.random.rand()
    if(this_state == "c"):
        if(u<=0.7):
            next_state = "c"
        else:
            next_state = "p"
    else:
        if(u<=0.5):
            next_state = "c"
        else:
            next_state = "p"
    return next_state

for i in range(5):
    this_stage = "c"
    result = []
    for j in range(9):
        result.append(this_stage)
        next_state = soda_simul(this_stage)
        this_stage = next_state

    print(''.join(result))
```

```
## ccppppccc
## cppccpppp
## ccccccccc
## cppppccpp
## cccpppppp
```

p.28 DTMC Simulator

R code

```r
cost_eval <- function(path) {
cost_one_path <-
str_count(path, pattern = "c")*1.5 +
str_count(path, pattern = "p")*1
return(cost_one_path)
}
MC_N <- 10000
spending_records <- rep(0, MC_N)
for (i in 1:MC_N) {
path <- "c" # coke today (day-0)
for (t in 1:9) {
this_state <- str_sub(path, -1, -1)
next_state <- soda_simul(this_state)
path <- paste0(path, next_state)
}
spending_records[i] <- cost_eval(path)
}
mean(spending_records)
```

Python code

```python
def cost_eval(path):
    cost_one_path = path.count("c")*1.5 + path.count("p")*1
    return cost_one_path

MC_N = 100
record = np.array([])
for i in range(MC_N):
    this_stage = "c"
    result = []
    for j in range(9):
        result.append(this_stage)
        next_state = soda_simul(this_stage)
        this_stage = next_state
```

```
    record = np.append(record, np.array([cost_eval(''.join(result))]))

print(np.mean(record))
```

```
## 12.065
```