C1_Jeong,wonryeol

Jeong, wonryeol

2021-01-04

25p

convert R to python

```
import numpy as np
def soda_simul(a):
  u = np.random.uniform(0,1,1)


  if a == "c":
    if u <= 0.7:
      return "c"
    else:
      return "p"
  else :
    if u <= 0.5:
      return "c"
    else:
      return "p"
```

Using the function soda_simul() let's generate 5 possible path for 10 days

```
for i in range(0,5):
  path = "c"
  for n in range(0,9):
    this_state=path[-1] # last element

    next_state=soda_simul(this_state)

    path=path+next_state
  print(path)
```

```
## ccccccccp
## cppccppccc
## ccpcccccpc
## cccppccccc
## cpccccpcp
```

- To address the second question regarding expected spending, we certainly need more than 5 paths
- Let's do it with 10,000 Monte-Carlo simulation
- we need cost evaluating function that calculates cost for each path

```
def cost_eval(path):

    cost_one_path=path.count("c")*1.5+path.count("p")*1
    return cost_one_path

MC_N=100000

spending_records=np.arange(0,MC_N)

for i in range(MC_N):
    path="c"
    for t in range (9):
        this_state = path[-1]
        next_state = soda_simul(this_state)
        path=path+next_state
    spending_records[i]=cost_eval(path)

np.mean(spending_records)
```

```
## 13.10171
```