이론통계학 HW2 - 포트폴리오 시뮬레이션

Jihye

2021년 3월 15일

#1. 최적 포트폴리오 찾기

a) a= 0.5 이고 Simulation 반복 횟수가 M=1000인 경우 목표달성확률을 구하시오.

1) 난수 생성하여 v0 ~ v20 matrix 만들기

```
n <- 20
M <-1000
mu <- 0.05; sigma <- 0.3
v0 <- 1; maturity <- 5
risk.free <- 0.02; target.return <- 0.04
a <- 0.5

v.matrix <- function(n,M,mu,sigma,v0,risk.free,a){
  set.seed(1234)
  random.normal <- matrix(rnorm(M*n,0,1), nrow = M, ncol = n)
  value.matrix <- matrix(1, nrow = M, ncol = n+1)
  colnames(value.matrix) <- c(0:n)
  value.matrix[,1] <- c(v0)
  rt <- exp(mu + sigma*random.normal) - 1
  for (i in 1:n){
    value.matrix[,i+1] <- value.matrix[,i]*(1 + (1-a)*risk.free + a*rt[,i])
  }
  return(value.matrix)
}
value.matrix <- v.matrix(n,M,mu,sigma,v0,risk.free,a)
head(value.matrix)
```

```
##      0         1         2         3         4         5         6
## [1,] 1 0.8759469 0.7674497 0.6926010 0.6775642 0.6123567 0.5635961
## [2,] 1 1.0812557 1.1735841 1.1972402 1.3448383 1.2396479 1.3553482
## [3,] 1 1.2377375 1.0412431 1.0604650 1.1522160 1.1589456 1.0533053
## [4,] 1 0.7700565 0.8824949 1.1133968 1.5839758 2.2871655 2.2693788
## [5,] 1 1.1078534 1.2840458 1.0655614 1.4468055 1.6765717 1.6020178
## [6,] 1 1.1218119 0.9050062 0.8091702 0.7946701 0.9641673 1.1454664
##             7         8         9        10        11        12        13
## [1,] 0.5857664 0.6306091 0.6246134 0.7012122 0.5713227 0.4677719 0.5035405
## [2,] 1.3611729 1.5209717 1.4928119 1.5471721 1.7706608 2.4887357 2.2297306
## [3,] 0.9413093 0.8953296 0.8132069 0.7876072 0.8852903 0.9206781 0.9064821
## [4,] 2.3563086 2.8951589 3.1901727 2.8456001 3.0115954 2.7309083 2.6314351
## [5,] 1.5851670 2.0706975 2.2403686 1.9159487 2.5364499 2.7848288 2.9930956
## [6,] 1.2162518 1.2281229 1.4493112 1.3818968 1.3311655 1.0593445 1.2328704
##            14        15        16        17        18        19       20
## [1,] 0.6145875 0.8244097 1.1127953 0.9952225 0.9502751 1.2679526 1.592555
## [2,] 2.2891089 2.7344100 2.6313270 2.3682767 2.5810937 2.4773520 2.640176
## [3,] 0.9418980 0.8460623 0.9430665 0.9520785 0.9597728 0.7122778 0.993350
## [4,] 3.3835560 3.5027036 4.1646274 4.3008912 7.7022014 9.8903461 8.897981
## [5,] 2.7100160 2.7467437 3.0852319 2.6481345 2.8251131 2.2818520 1.936648
## [6,] 1.2259018 1.2748205 1.0141169 1.1069497 0.9824012 1.2430050 1.398717
```

2) 목표달성한 횟수 계산

```
rsfc <- function(n,M,mu,sigma,v0,risk.free,a,target.return){
  target.matrix <- matrix(v0*((1+target.return)^c(1:n)),nrow = M, ncol = n, byrow=T)
  v.matrix <- v.matrix(n,M,mu,sigma,v0,risk.free,a)
  rsfc <- colMeans(v.matrix[,-1]>target.matrix)
  return(rsfc)
}
rsfc(n,M,mu,sigma,v0,risk.free,a,target.return)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## 0.470 0.497 0.514 0.518 0.534 0.542 0.544 0.567 0.568 0.568 0.571 0.561
##    13    14    15    16    17    18    19    20
## 0.571 0.570 0.584 0.572 0.569 0.566 0.565 0.568
```

b) 처음 5번의 Simulation에 대하여 Vt의 시계열도표를 겹쳐서 그려보시오.

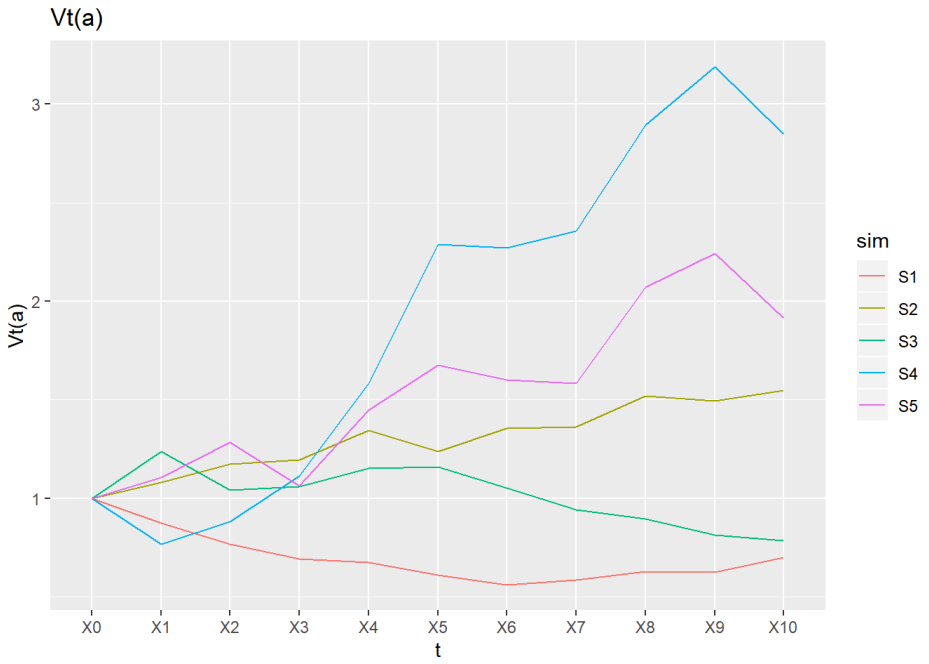
```
sim5 <- data.frame(value.matrix[c(1:5),c(1:11)])
sim5$sim <-  c("S1","S2","S3","S4","S5")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

```
## √ ggplot2 3.2.1     √ purrr   0.3.3
## √ tibble  2.1.3     √ dplyr   0.8.3
## √ tidyr   1.0.0     √ stringr 1.4.0
## √ readr   1.3.1     √ forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
new_sim <- gather(sim5, type, value, -sim)
ggplot(new_sim, aes(x=type, y=value, group = sim, color = sim)) +
  geom_line() +
  scale_x_discrete(limit = c("X0","X1","X2","X3","X4","X5","X6","X7","X8","X9","X10")) +
  labs(x="t", y = "Vt(a)", title = "Vt(a)")
```



a = 0, 0.1,..., 1일 때 목표달성확률을 구하고 이 값들의 그래프를 그리시오.

```
a_rsfc <- data.frame(a = seq(0,1,0.1), rsfc = NA)
for(i in 0:10){
  a_rsfc[i+1,2] <- rsfc(n,M,mu,sigma,v0,risk.free,a_rsfc[i+1,1],target.return)[maturity]
}
a_rsfc
```

```
##      a  rsfc
## 1  0.0 0.000
## 2  0.1 0.188
## 3  0.2 0.393
## 4  0.3 0.479
## 5  0.4 0.516
## 6  0.5 0.534
## 7  0.6 0.540
## 8  0.7 0.543
## 9  0.8 0.545
## 10 0.9 0.540
## 11 1.0 0.539
```

d) 목표달성확률을 최대로 하는 a값과 해당 목표달성확률을 구하시오.

```
a_rsfc[which.max(a_rsfc$rsfc),]
```

```
##     a  rsfc
## 9 0.8 0.545
```

e) (Sensitivity Analysis) 위 방법을 이용하여 목표수익률 r와 만기 *n*이 아래와 같이 주어질 경우 때 최적 *Portfolio a* 및 예상되는 목표달성확률을 구하는 표를 완성하시오.

```
rn.table <- data.frame("r.star"=c(0.03,0.03,0.04,0.04,0.06,0.06), "n" = rep(c(5,10),3), "a.star" = NA, "rsfc" = NA)
rn.table
```

```
##   r.star  n a.star rsfc
## 1   0.03  5     NA   NA
## 2   0.03 10     NA   NA
## 3   0.04  5     NA   NA
## 4   0.04 10     NA   NA
## 5   0.06  5     NA   NA
## 6   0.06 10     NA   NA
```

```
sensitivity<-function(n,M,mu,sigma,v0,risk.free,target.return,maturity){
  a_rsfc <- data.frame(a = seq(0,1,0.1), rsfc = NA)
  for(i in 0:10){
    a_rsfc[i+1,2] <- rsfc(n,M,mu,sigma,v0,risk.free,a_rsfc[i+1,1],target.return)[maturity]
  }
  return(a_rsfc[which.max(a_rsfc$rsfc),])
}
for(i in c(1:nrow(rn.table))){
  rn.table[i,c(3,4)] <- sensitivity(n,M,mu,sigma,v0,risk.free,target.return = rn.table[i,1],maturity=rn.table[i,2])
}
rn.table
```

```
##   r.star  n a.star  rsfc
## 1   0.03  5    0.5 0.592
## 2   0.03 10    0.4 0.638
## 3   0.04  5    0.8 0.545
## 4   0.04 10    0.7 0.584
## 5   0.06  5    1.0 0.482
## 6   0.06 10    0.9 0.489
```

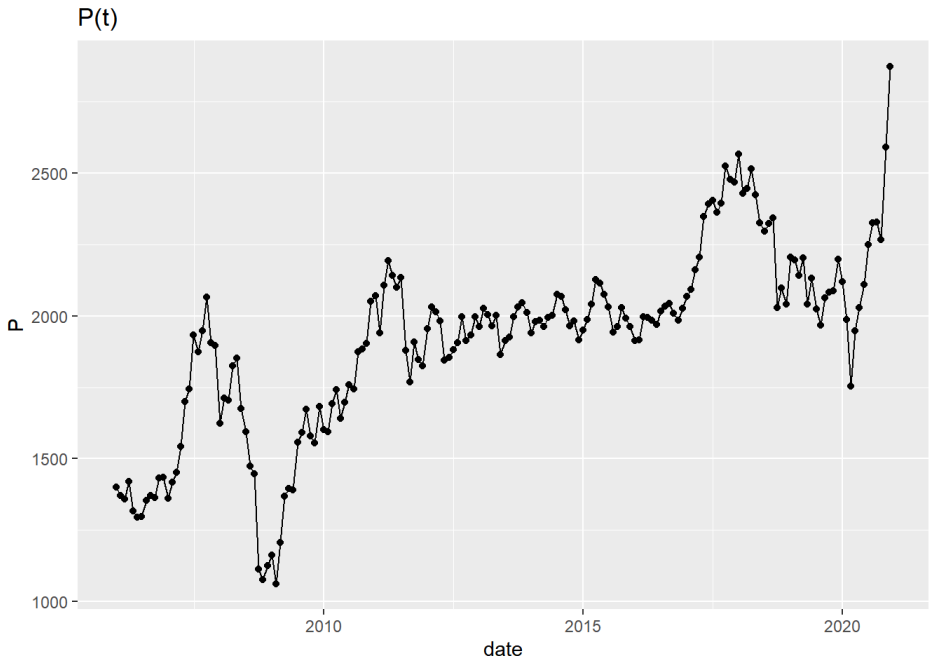
#2. Stock Price Model

```
setwd("C:/Users/JIHYE/Desktop/대학원/이론통계/과제2")
kospi <- read.csv("kospi.csv", skip = 3)
colnames(kospi) <- c("date", "P")
kospi<-kospi[1:180,]
kospi$P <- as.numeric(gsub(",","",kospi$P))
kospi$date <- as.Date(as.POSIXlt(paste(kospi$date,"01",sep = "/"),format = "%Y/%m/%d"))
head(kospi)
```

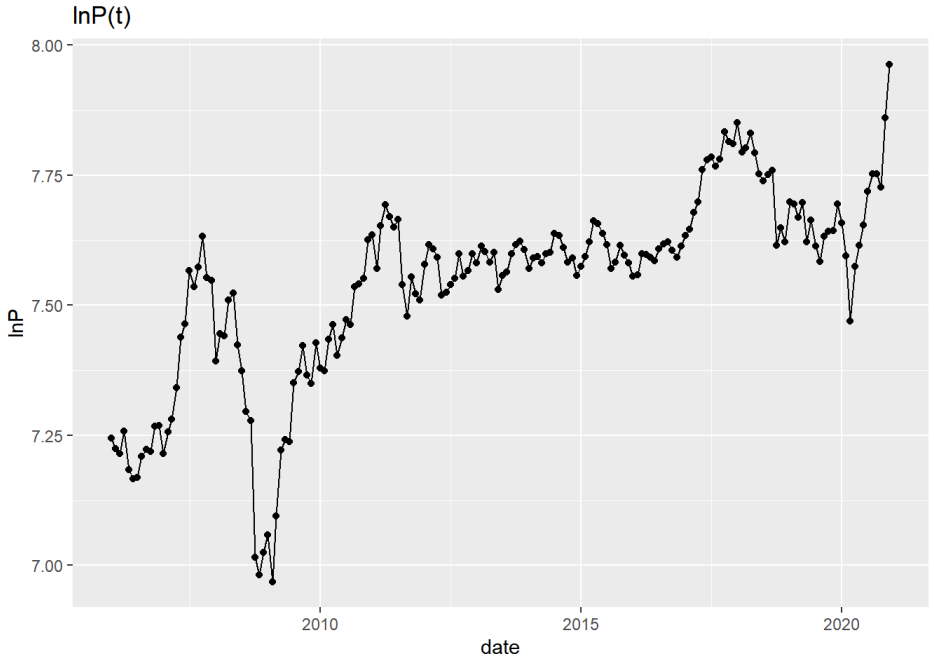
```
##         date       P
## 1 2006-01-01 1399.83
## 2 2006-02-01 1371.59
## 3 2006-03-01 1359.60
## 4 2006-04-01 1419.73
## 5 2006-05-01 1317.70
## 6 2006-06-01 1295.15
```

2006.1-2021.1 기간 월별 KOSPI 주가지수자료에 대해 P(t) 및 lnP(t)의 시계열도표(Time Series Plot)를 각 각 그리시오

```
kospi$lnP <- log(kospi$P)
ggplot(kospi, aes(date, P))+geom_point()+geom_line()+labs(title="P(t)")
```

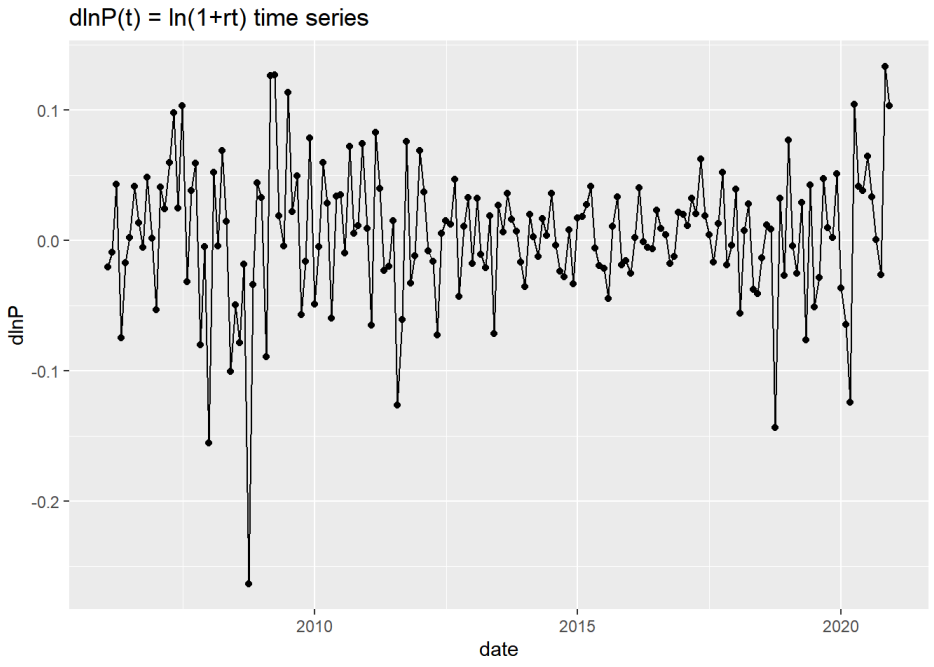



```
ggplot(kospi, aes(date, lnP))+geom_point()+geom_line()+labs(title="lnP(t)")
```



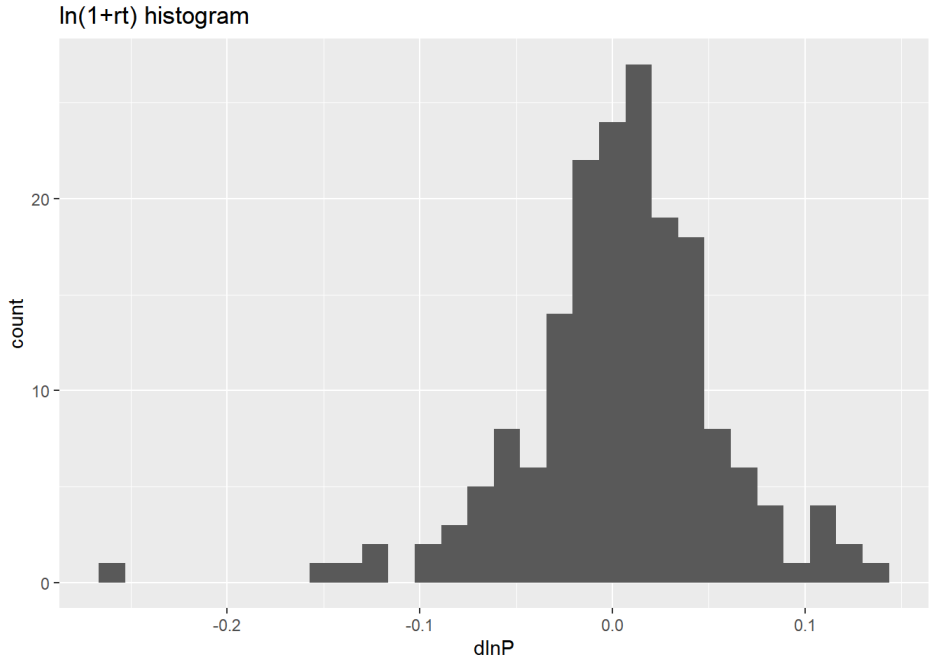
b) 월별 로그주가지수 lnP(t)의 차분을 계산하고 로그수익률 시계열도표 및 히스토그램을 그려보시오.

```
kospi$dlnP[1] <- NA
for (i in c(2:nrow(kospi))){
  kospi$dlnP[i] <- kospi$lnP[i] - kospi$lnP[i-1]
}
ggplot(kospi[-1,], aes(date, dlnP)) + geom_point() + geom_line() +
  labs(title="dlnP(t) = ln(1+rt) time series")
```



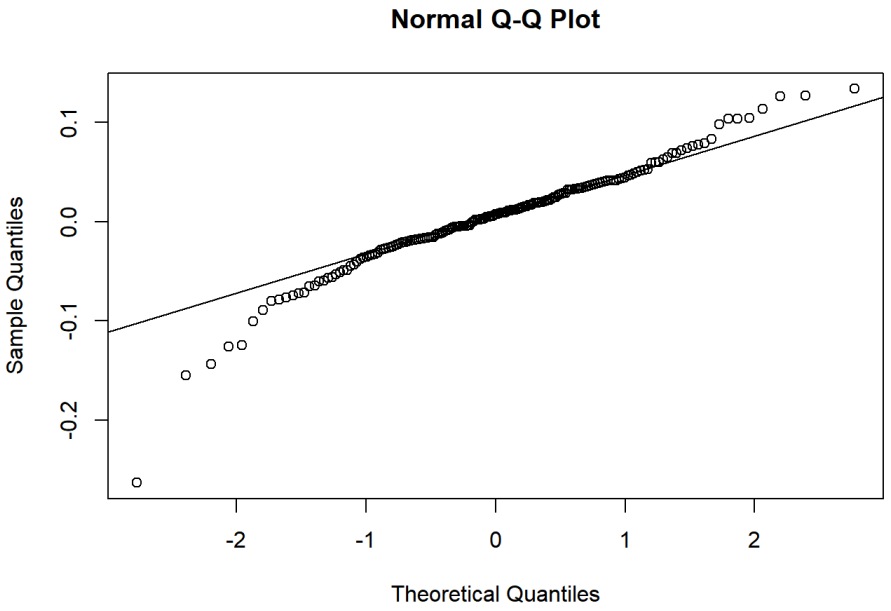

```
ggplot(kospi[-1,], aes(dlnP)) + geom_histogram() + labs(title="ln(1+rt) histogram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



c) 정규성 검정

```
qqnorm(kospi$dlnP);qqline(kospi$dlnP)
```



꼬리를 제외하고는 정규성을 거의 따른다고 할 수 있다.

d) (mu, sigma) 추정

```
dt <- 1/12
mu <- mean(kospi$dlnP, na.rm = T)/dt; mu
```

```
## [1] 0.04821248
```

```
sigma <- sd(kospi$dlnP, na.rm = T)/sqrt(dt); sigma
```

```
## [1] 0.1808074
```

e) 만기 n=10년, 목표수익률 r=0.04인 경우 최적 포트폴리오

```
sensitivity(n,M,mu=mu,sigma=sigma,v0,risk.free,target.return=0.04,maturity=10)
```

```
##    a  rsfc
## 11 1 0.582
```

f) 표 완성

```
rn.table <- data.frame("r.star" = rep(c(0.04,0.05), each=3), "n" = rep(c(5,10,20),2), "a.star" = NA, "rsfc" = NA)

for(i in c(1:nrow(rn.table))){
  rn.table[i,c(3,4)] <- sensitivity(n,M,mu,sigma,v0,risk.free,target.return = rn.table[i,1],maturity=rn.table[i,2])
}
rn.table
```

```
##   r.star  n a.star  rsfc
## 1   0.04  5      1 0.551
## 2   0.04 10      1 0.582
## 3   0.04 20      1 0.574
## 4   0.05  5      1 0.503
## 5   0.05 10      1 0.518
## 6   0.05 20      1 0.489
```