Solution of Week 3 Homework

Jingyu He

10/18/2018

Problem 1

Build Efficient Frontier

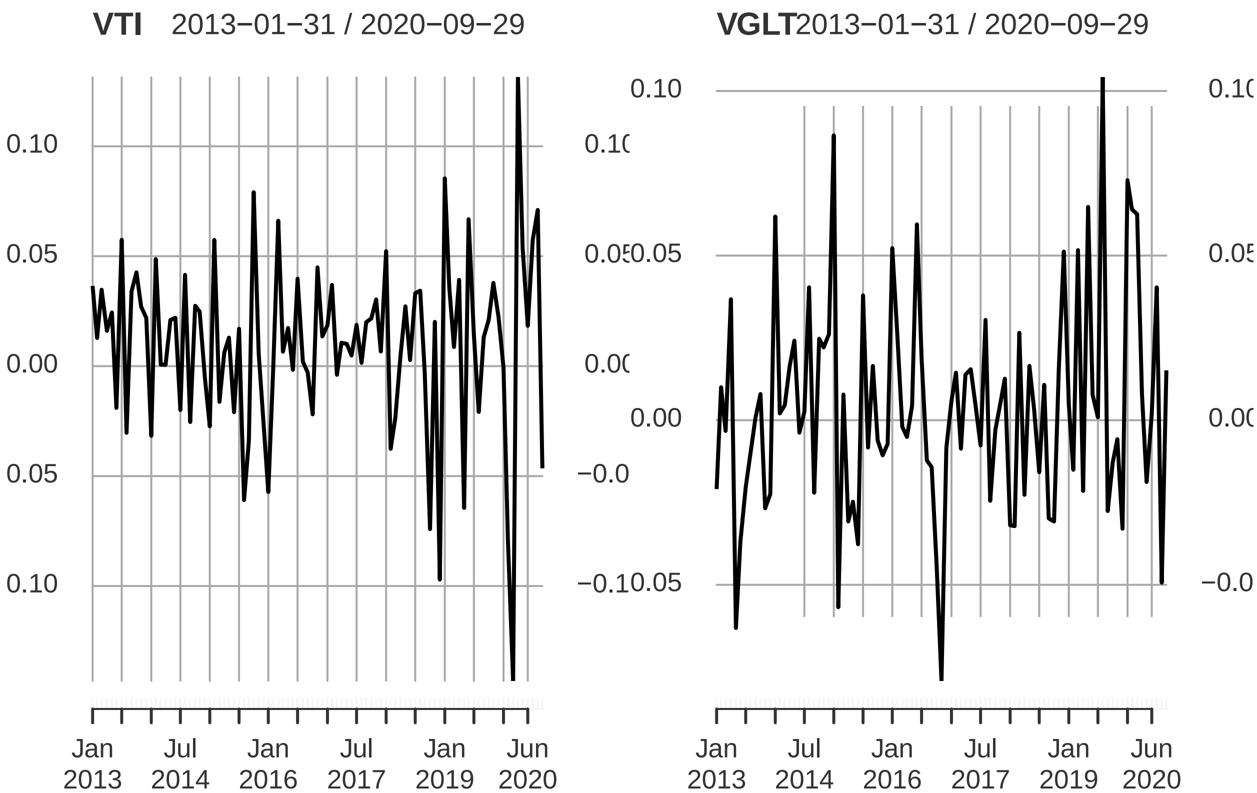
First we use package quantmod to download data from Yahoo finance and compute monthly returns.

```
# download data from yahoo finance by package quantmod
library(quantmod)
options("getSymbols.yahoo.warning"=FALSE)
# download ticker VTI, start from 2013/01/01 to now
getSymbols('VTI', from = '2013-01-01', warnings=FALSE)
# convert to monthly data
VTI = monthlyReturn(VTI)

# download VGLT
getSymbols('VGLT', from = '2013-01-01', warnings=FALSE)
VGLT = monthlyReturn(VGLT)
```

Take a look to the two time series

```
# plot two time series
par(mfrow = c(1,2))
plot(VTI)
plot(VGLT)
```



Calculate mean and standard deviation for two ETFs

```
# calculate expected return, standard error and covariance of two assets
mean_VTI = mean(VTI)
mean_VGLT = mean(VGLT)
sd_VTI = sd(VTI)
sd_VGLT = sd(VGLT)
cov_VTI_VGLT = as.numeric(cov(VTI, VGLT))
```

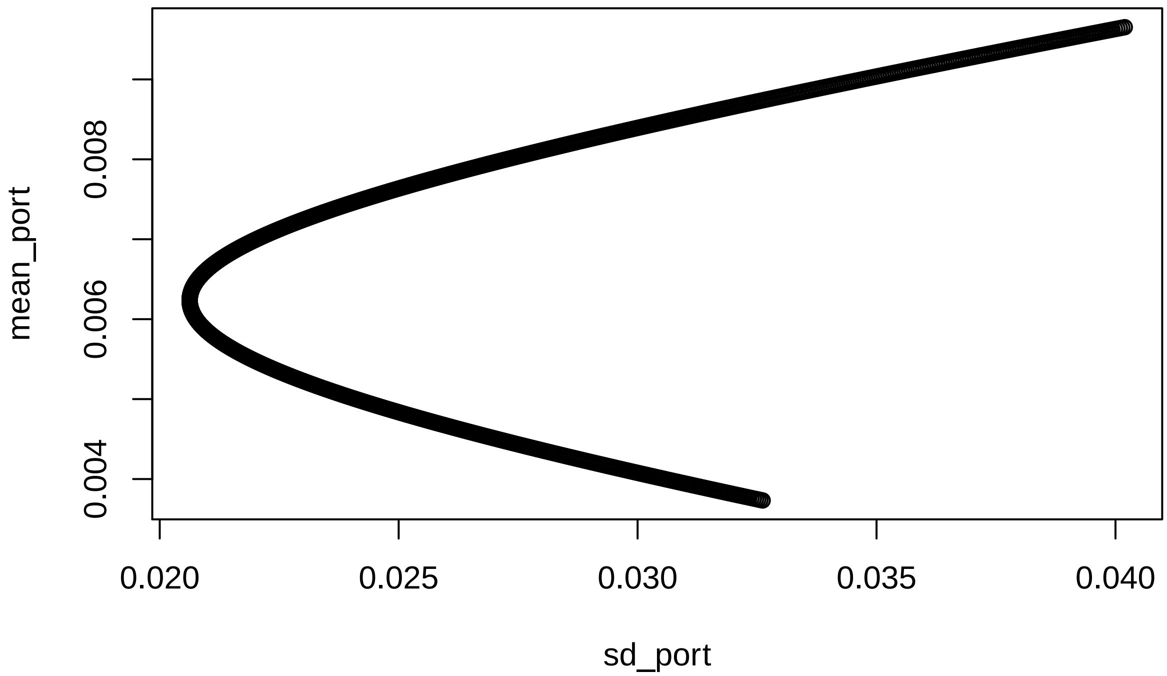
Create different portfolios and calculate their expected return and standard deviations

```
# create 1000 synthetic portfolio weights
weights = seq(from = 0, to = 1, length.out = 1000)

# create corresponding returns and standard deviation of all 1000 portfolios
mean_port = weights * mean_VTI + (1 - weights) * mean_VGLT
sd_port = sqrt(weights^2 * sd_VTI^2 + (1 - weights)^2 * sd_VGLT^2 + 2 * weights * (1 - weights) * cov_V
```

Draw the efficient frontier

```
# plot efficient frontier
plot(sd_port, mean_port)
```



##

Find the allocation with best Sharp ratio

```
# compute sharp ratio
sharp = mean_port / sd_port
# find index of biggest sharp ratio
ind = which.max(sharp)
# corresponding weights
weights[ind]
```

```
## [1] 0.5355355
```

50-50 allocation

```
# half and halp portfolio
mean_port2 = 0.5 * mean_VTI + 0.5 * mean_VGLT
sd_port2 = sqrt(0.5^2 * sd_VTI^2 + 0.5^2 * sd_VGLT^2 + 2 * 0.5 * (1 - 0.5) * cov_VTI_VGLT)

# this function gives probability return > 0.01
# if lower.tail = FALSE, give P(return > 0.01)
# if lower.tail = TRUE, give P(return < 0.01)
pnorm(0.01, mean = mean_port2, sd = sd_port2, lower.tail =  FALSE)
```

```
## [1] 0.4378291
```

Problem 2

Invest in S&P 500, initial investment $100,000

Suppose we invest in SPY, which is the biggest ETF of S&P 500 index. Similar to problem 1, first we download data and compute monthly returns and historical mean and standard deviation.

```
set.seed(100)

# download ticker VTI, start from 2013/01/01 to now
getSymbols('SPY', from = '2013-01-01', warnings=FALSE)
# convert to monthly data
SPY = monthlyReturn(SPY)
```

Compute mean and standard deviation

```
mean_SPY = mean(SPY)
sd_SPY = sd(SPY)
mean_SPY
```

```
## [1] 0.009699301
```

```
sd_SPY
```

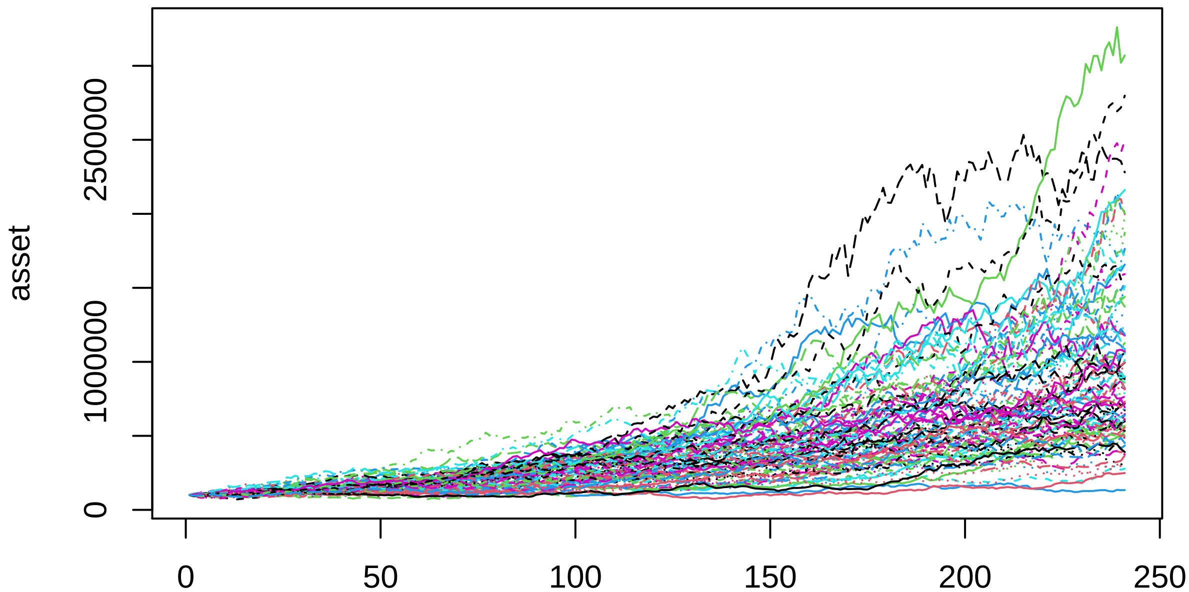
```
## [1] 0.038727
```

Suppose future monthly return follows normal distribution with the same mean and standard deviation, we can run simulations to 20 years (240 monthes). Repeat the simulation 100 times.

```
M = 100 # repeat 100 simulations
T = 240 # time lenght, 20 years = 240 months
asset = matrix(0, T+1, M)
asset[1,] = 100000 # initial value
for(m in 1:M){
  for(i in 1:240){
    asset[i + 1, m] = asset[i, m] * (1 + rnorm(1, mean = mean_SPY, sd = sd_SPY))
  }
}
```

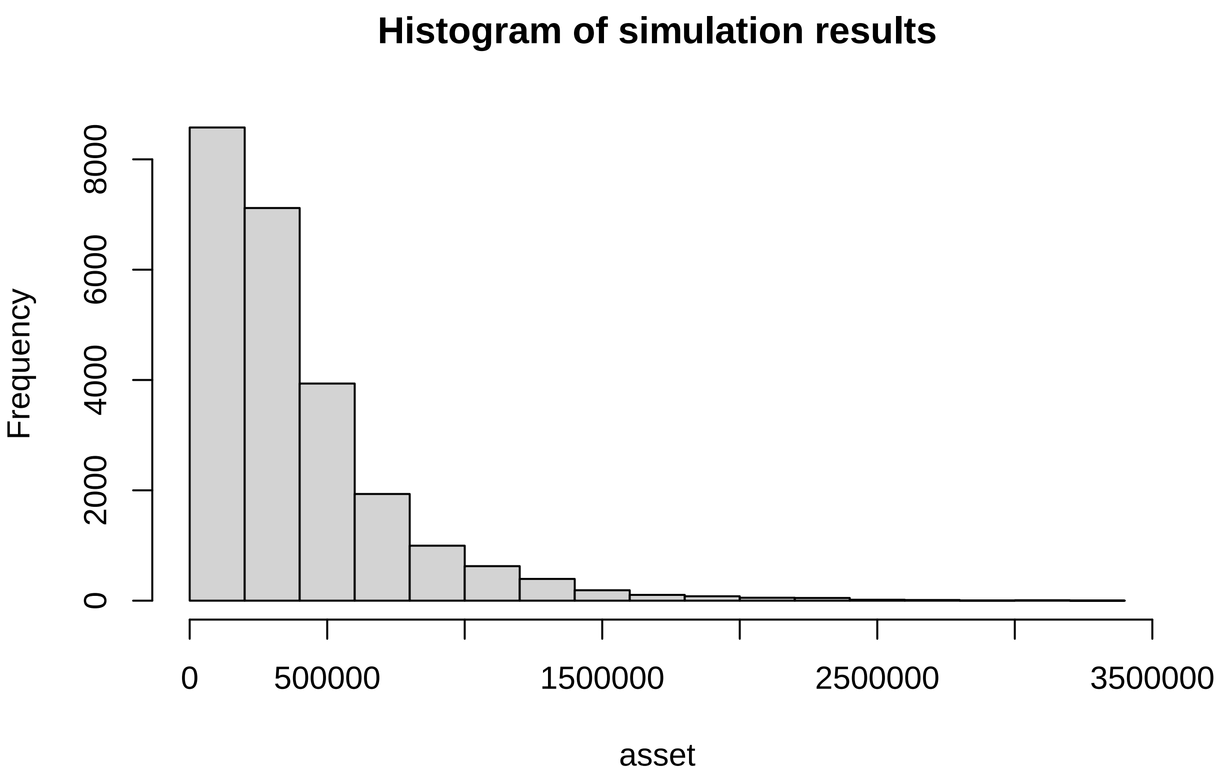
Let's plot the chart of 100 simulations

```
matplot(asset, type = "l")
```



Plot the distribution of asset in the last month

```
endvalue = asset[T+1,]
hist(asset, main = "Histogram of simulation results")
```



On average, the money I can get after 20 years

```
mean(endvalue)
```

```
## [1] 977421.9
```

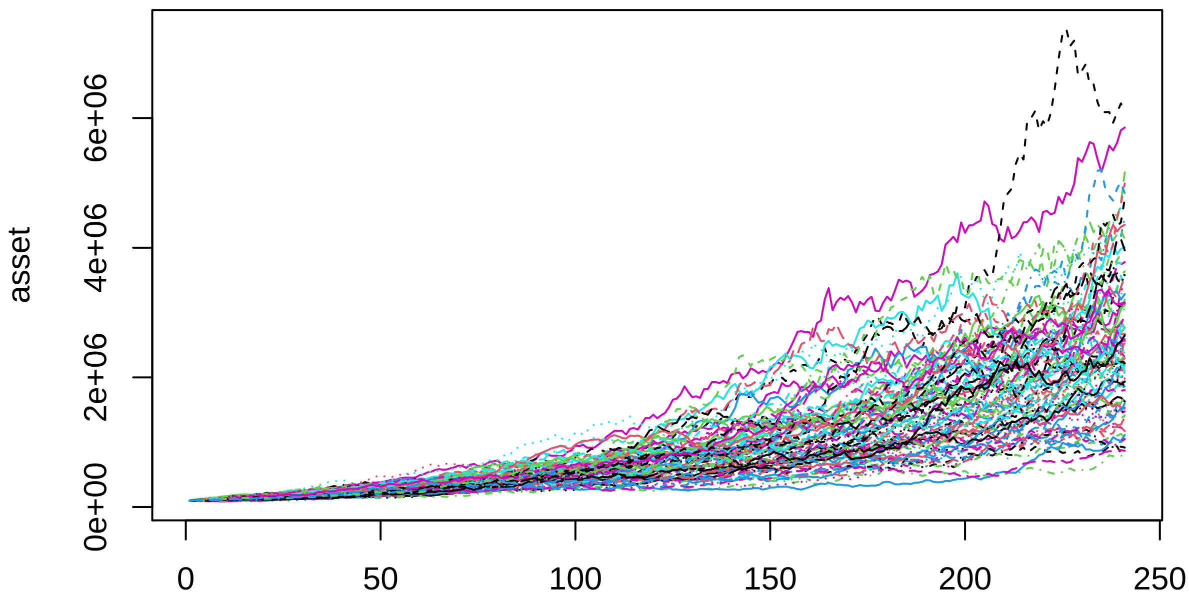
Invest in S&P 500, initial investment $100,000, extra $20,000 a year

Suppose we invest extra 20000/12 a month, repeat previous analysis

```
M = 100 # repeat 100 simulations
T = 240 # time lenght, 20 years = 240 months
asset = matrix(0, T+1, M)
asset[1,] = 100000 # initial value
for(m in 1:M){
  for(i in 1:240){
    asset[i + 1, m] = asset[i, m] * (1 + rnorm(1, mean = mean_SPY, sd = sd_SPY)) + 20000/12
  }
}
```

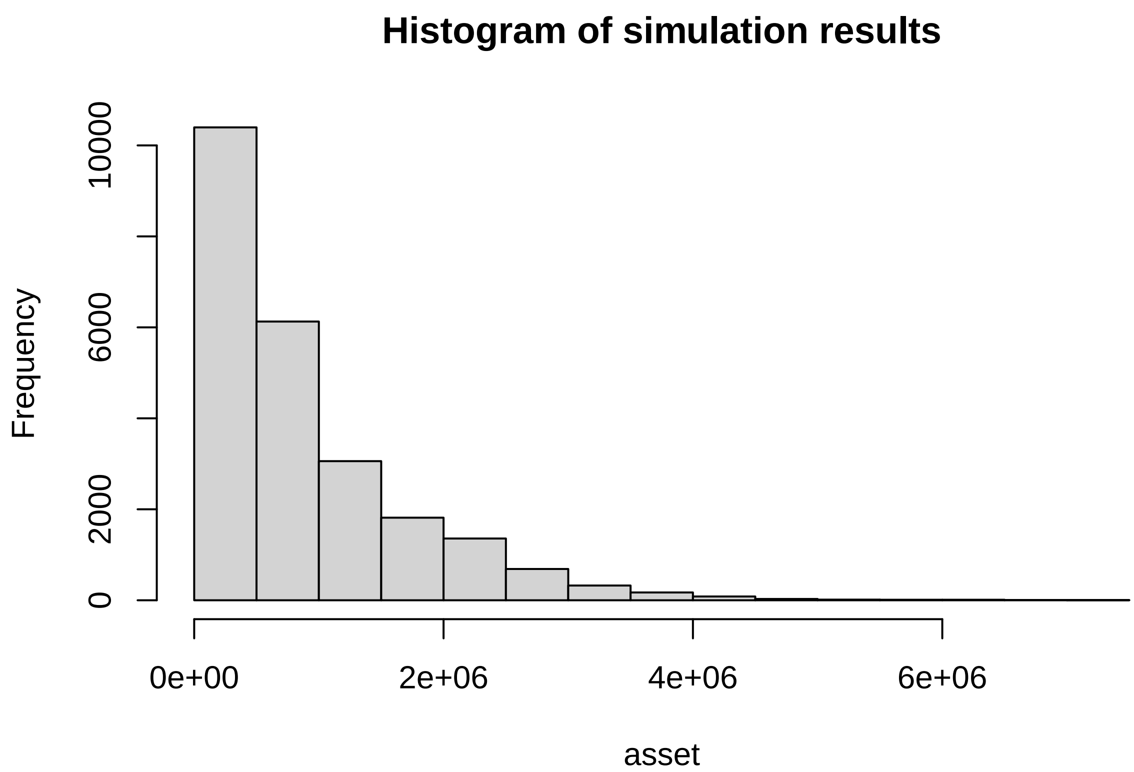
Let's plot the chart of 100 simulations

```
matplot(asset, type = "l")
```



Plot the distribution of asset in the last month

```
endvalue = asset[T+1,]
hist(asset, main = "Histogram of simulation results")
```



On average, the money I can get after 20 years

```r
mean(endvalue)
```

```
## [1] 2537652
```

Invest in S&P 500 and U.S. treasury

We invest in TLT, which is ETF of long term U.S. treasury. First download data of TLT

```
# download ticker VTI, start from 2013/01/01 to now
getSymbols('TLT', from = '2013-01-01', warnings=FALSE)
# convert to monthly data
TLT = monthlyReturn(TLT)
```

Calculate mean and standard deviation

```
mean_TLT = mean(TLT)
sd_TLT = sd(TLT)
mean_TLT
```

```
## [1] 0.004031027
```

```
sd_TLT
```

```
## [1] 0.03475186
```

Calculate covariance of SPY and TLT, then compute mean and standard deviation of a 50-50 portfolio

```
cov_SPY_TLT = as.numeric(cov(SPY, TLT))
mean_port = 0.5 * mean_SPY + 0.5 * mean_TLT
sd_port = sqrt(0.5^2 * sd_TLT^2 + 0.5^2 * sd_SPY^2 + 2 * 0.5 * (1 - 0.5) * cov_SPY_TLT)
```

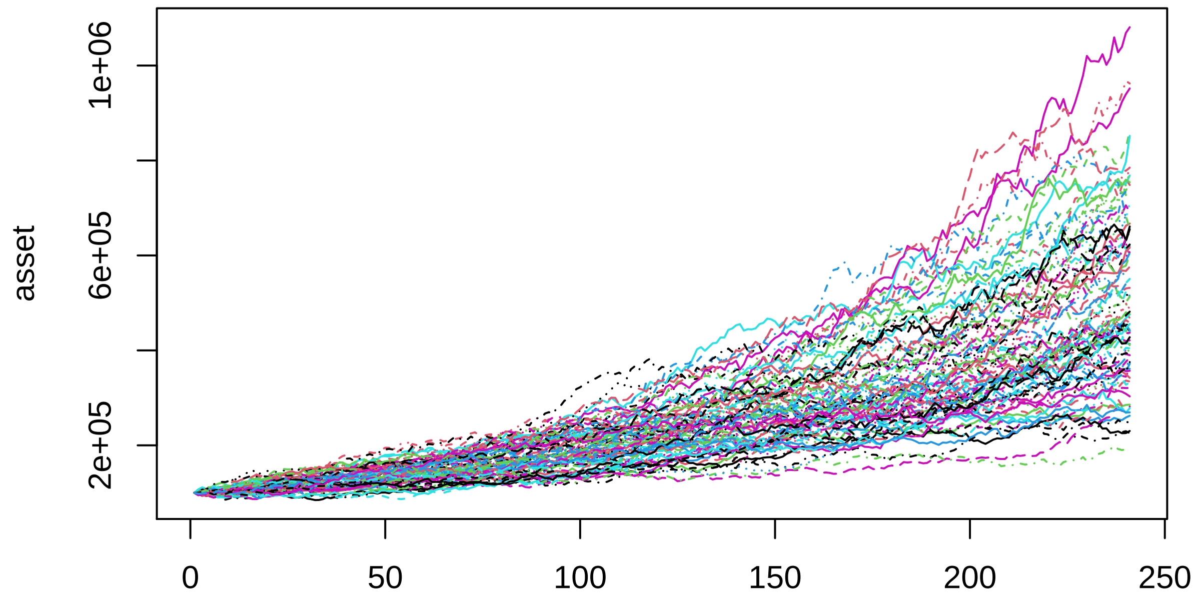
Repeat simulations above

Initival investment $100,000

```
M = 100 # repeat 100 simulations
T = 240 # time lenght, 20 years = 240 months
asset = matrix(0, T+1, M)
asset[1,] = 100000 # initial value
for(m in 1:M){
  for(i in 1:240){
    asset[i + 1, m] = asset[i, m] * (1 + rnorm(1, mean = mean_port, sd = sd_port))
  }
}
```

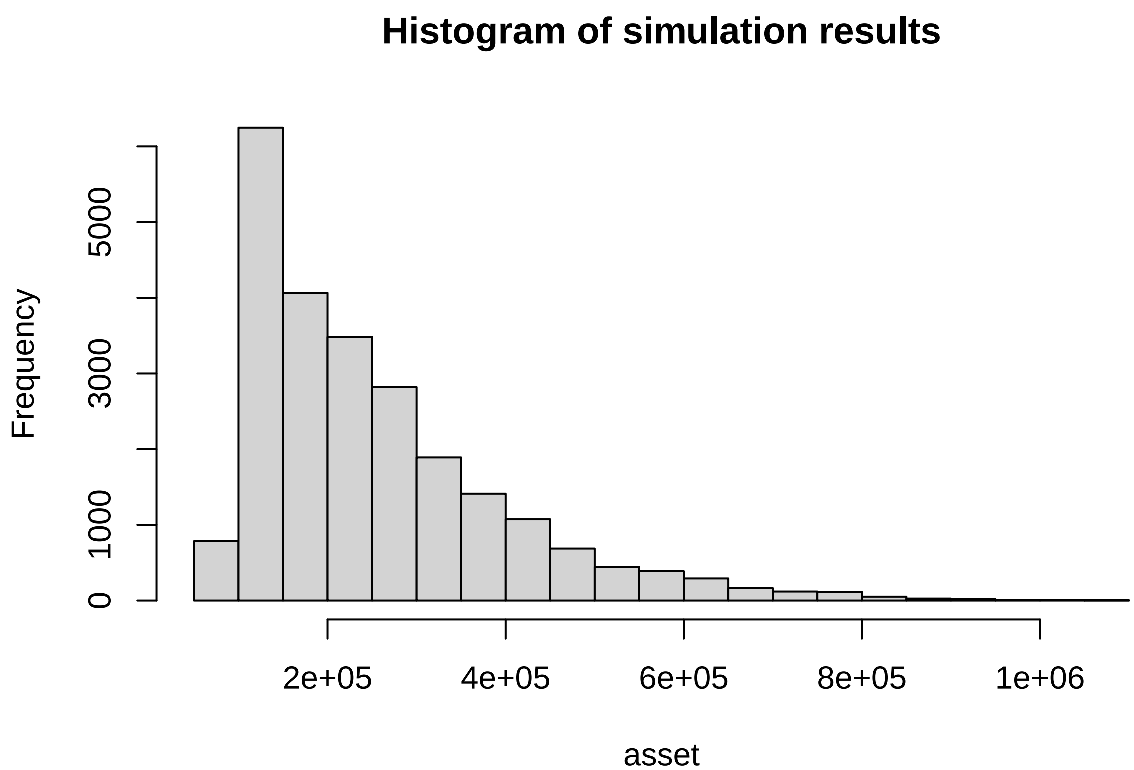
Let's plot the chart of 100 simulations

```
matplot(asset, type = "l")
```



Plot the distribution of asset in the last month

```
endvalue = asset[T+1,]
hist(asset, main = "Histogram of simulation results")
```



On average, the money I can get after 20 years

```
mean(endvalue)
```

```
## [1] 511138.1
```

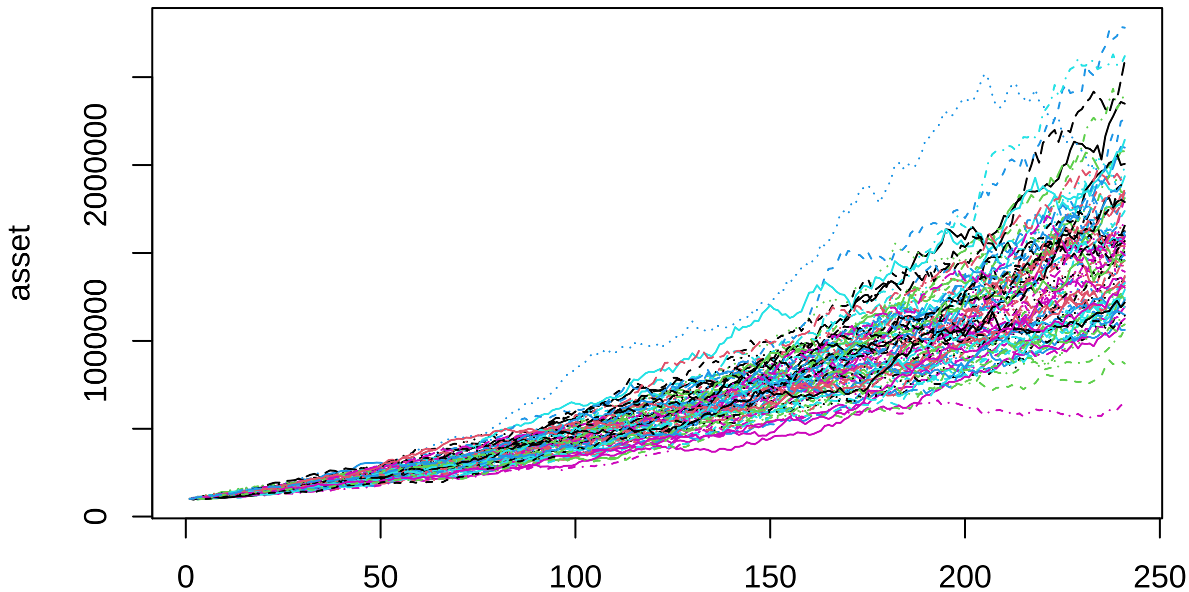
Initial investment $100,000, extra $20,000 a year

Suppose we invest extra 20000/12 a month, repeat previous analysis

```
M = 100 # repeat 100 simulations
T = 240 # time lenght, 20 years = 240 months
asset = matrix(0, T+1, M)
asset[1,] = 100000 # initial value
for(m in 1:M){
  for(i in 1:240){
    asset[i + 1, m] = asset[i, m] * (1 + rnorm(1, mean = mean_port, sd = sd_port)) + 20000/12
  }
}
```

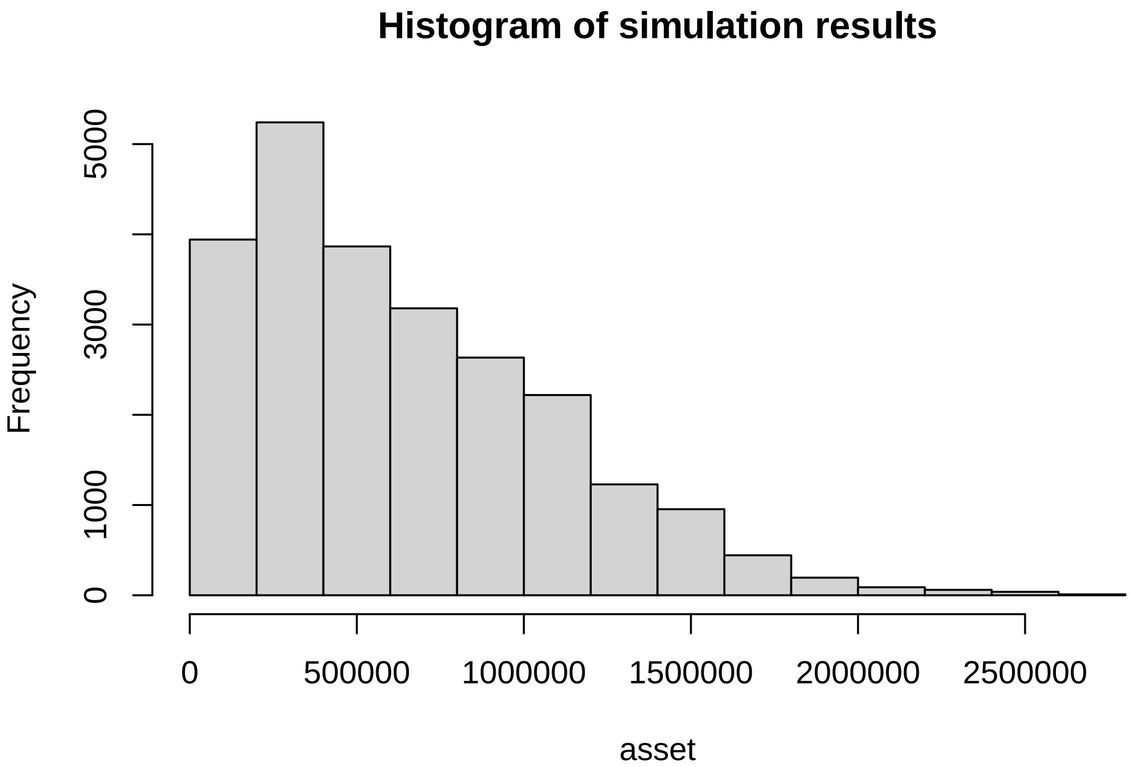
Let's plot the chart of 100 simulations

```
matplot(asset, type = "l")
```



Plot the distribution of asset in the last month

```
endvalue = asset[T+1,]
hist(asset, main = "Histogram of simulation results")
```



On average, the money I can get after 20 years

```
mean(endvalue)
```

```
## [1] 1561395
```