# MTH9855 Asset Allocation and Portfolio Management

- HW 01
- **Author: Hongchao Pan**
- Kernel Version: R 3.3.2

In [1]:
```
# Load the library
library(repr)
```

## Problem 1.1

- In probability theory, if X is a random variable, then we have Jensen's inequality stats that $\phi(\mathbb{E}(X)) \leq \mathbb{E}(\phi(X))$ if and only if $\phi$ is a convex function. Or we can state this as: $\mathbb{E}(\phi(X)) \leq \phi(\mathbb{E}(X))$ if and only if $\phi$ is a concave function.

- Let $\phi \stackrel{d}{=} u, X \stackrel{d}{=} w + \tilde{z}$, then we have:
$$\mathbb{E}(u(w + \tilde{z})) \leq u(\mathbb{E}(w + \tilde{z})) = u(w + \mathbb{E}(\tilde{z}))$$

if and only if $u$ is a concave function.

## Problem 1.2

- From Lecture 1, we have the risk premium defined by:

$$\mathbb{E}(u(w+\tilde{z})) = u(w-\Pi)$$

- Plug the utility function into above equation, we have:

$$\frac{-\exp(-a(w-\Pi))}{a} = \frac{-\mathbb{E}[\exp(-a(w-\Pi))]}{a}$$

$\Rightarrow$

$$\begin{aligned}\Pi &= w + \frac{\log[\mathbb{E}(\exp(-a(w+\tilde{z}))]}{a} \\ &= w + \frac{1}{a}\log\exp(-aw) + \frac{1}{a}\log\exp(-a\tilde{z}) \\ &= w - w + \frac{1}{2}a\sigma^2 \\ &= \frac{1}{2}\sigma^2 A(w)\end{aligned}$$

As $A(w) = -\frac{u''}{u'} = a$

- With given condition in the question, $u(w) = -\exp(-aw)/a$, we can compute the $\mathbb{E}[u(\tilde{w})]$ as:

$$\begin{aligned}\mathbb{E}\left[u(\tilde{w})\right] &= \int_{-\infty}^{\infty} -\frac{1}{a}e^{-a\tilde{w}}\,\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(\tilde{w}-\mu)^2}{2\sigma^2}}\,d\tilde{w} \\ &= -\frac{1}{a}\int_{-\infty}^{\infty}\frac{1}{\sigma\sqrt{2\pi}}e^{(\tilde{w}-(\mu-a\sigma))^2}e^{-\frac{\mu^2-(\mu-a\sigma^2)^2}{2\sigma^2}}\,d(\tilde{w}-(\mu-a\sigma))^2) \\ &= -\frac{1}{a}e^{-\frac{\mu^2-(\mu-a\sigma^2)^2}{2\sigma^2}} \\ &= -\frac{1}{a}e^{-a(\mu-\frac{1}{2}a\sigma^2)} \\ &= u(\mu - \frac{1}{2}a\sigma^2)\end{aligned}$$

## Problem 1.3

- From lecture 1, if agent $v$ is more risk-averse than agent $u$, we have: $\Pi_v \geq \Pi_u$.
- From equation (1) in lecture 1, we know: $\Pi \sim \frac{1}{2}\sigma^2 A(w)$. Then we have: $A_v(w) \geq A_u(w)$, i.e., (a) and (b) are equivalent.
- Given $v(w) = \phi(u(w))$. As $v$ and $u$ are strictly increasing function, we have: $u' > 0, v' > 0$.
- $v(w) = \phi(u(w)) \to v' = \phi'(u)u' > 0 \to \phi' > 0$.
- Given the eqaution, we can compute the $A(w)$ of agent $v$ and ageng $u$ as:

$$\begin{aligned} A_v(w) &= \frac{-v''}{v'} \\ &= -\frac{\phi''(u')^2 + \phi' u''}{\phi' u'} \\ &= -\frac{\phi'' u'}{\phi'} - \frac{u''}{u'} \\ &= -\frac{\phi'' u'}{\phi'} + A_u(w) \end{aligned}$$

$\Rightarrow$

$$A_v(w) - A_u(w) = -\frac{\phi'' u'}{\phi'}$$

- If (a) or (b) holds, we have $A_v(w) \geq A_u(w)$, then we have $\phi'' < 0$ with $\phi' > 0$

- Then we have (a), (b), and (c) are equivalent.

## Problem 1.4

- From lecture 1, we have the risk premium is defined by:

$$\mathbb{E}[u(w+\tilde{z})] = u(w-\Pi)$$

- Then for function $v$, we can compute the risk premium as:

$$\begin{aligned}\mathbb{E}[v(w+\tilde{z})] &= \mathbb{E}[a+bu(w+\tilde{z})]\\ &= a+b\mathbb{E}[u(w+\tilde{z})]\\ &= a+bu(w-\Pi)\\ &= v(w-\Pi)\end{aligned}$$

- This indicates that $v$ and $u$ make the same decisions with same risk $\tilde{z}$ and same risk premium $\Pi$.

## Problem 1.5

**Answer of (a):**

- With $h'r = \sum_i h_i r_i$, we have:

$$\begin{aligned}\mathbb{E}[h'r] &= \frac{1}{n}\sum_i \mathbb{E}[r_i]\\ &= \beta\mathbb{E}[r_M] + \frac{1}{n}\sum_i E[\epsilon_i]\end{aligned}$$

$$\begin{aligned}\mathbb{V}[h' r] &= \mathbb{V}[\sum_i \frac{1}{n}(\beta r_M + \epsilon_i)] \\ &= \mathbb{V}[\beta r_M + \frac{1}{n}\sum_i \epsilon_i] \\ &= \beta^2 \sigma_M^2 + \mathbb{V}[\frac{1}{n}\sum_i \epsilon_i] \\ &= \beta^2 \sigma_M^2 + \frac{1}{n^2}\sum_i \sigma_i^2\end{aligned}$$

- As $\mathbb{V}(h' r) = f(\beta, \sigma_M^2) + g(\sigma_1^2, \ldots, \sigma_n^2)$, we then have:

$$f = \beta^2 \sigma_M^2$$
$$g = \frac{1}{n^2}\sum_i \sigma_i^2$$

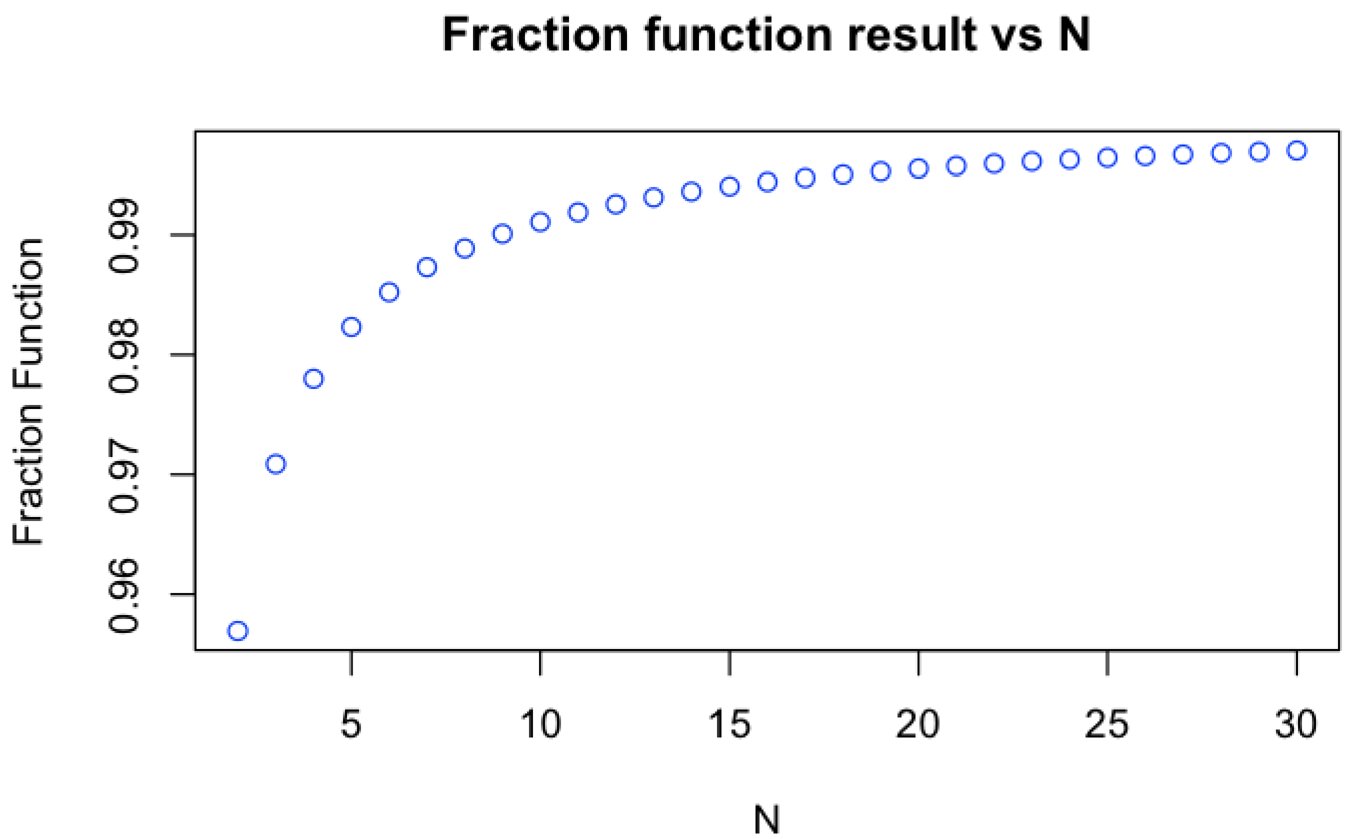
**Answer of (b):**

In [2]:
```
# Define a function to compute the fraction f/(f+g)
beta <- 0.5
sigma_M <- 0.2
sigma_i <- 0.03

fracFunc <- function(n){
    # Only input is n
    f <- beta^2 * sigma_M^2
    g <- (n* sigma_i^2)*1.0/(n^2)

    return (f*1.0/(f+g))
}
```

In [3]:
```
# Compute the fraction function
N <- 2:30
fracRes <- fracFunc(N)
fracRes
```

0.956937799043062 0.970873786407767 0.97799511002445 0.982318271119843 0.985221674876847
0.987306064880113 0.988875154511743 0.99009900990099 0.991080277502478 0.991884580703336
0.992555831265509 0.993124522536287 0.99361249112846 0.99403578528827 0.994406463642014
0.994733762434172 0.995024875621891 0.995285489785228 0.995520159283225 0.995732574679943
0.995925758261657 0.996102208748376 0.99626400996264 0.996412913511359 0.996550402453047
0.996677740863787 0.996796012815949 0.996906153317291 0.997008973080758

In [4]:
```
# Plot the fraction w.r.t. to N
options(repr.plot.height=4,repr.plot.width=6)
plot(N,fracRes,col="blue",xlab="N",ylab="Fraction Function",
    main="Fraction function result vs N")
```




**Answer of (c):**

- With $\mathbb{E}[r_M] = 0.07, \mathbb{E}[\epsilon_i] = 1.5\sigma_i$ or $\mathbb{E}[\epsilon_i] = 2\sigma_i$ , we have:

In [5]:
```
# Define a function to compute the sharpe ratio
beta <- 0.5
sigma_M <- 0.2
sigma_i <- 0.03
E_rm <- 0.07
E_ei1 <- 1.5*sigma_i
E_ei2 <- 2.0*sigma_i

SharpeRatio <- function(n,E_ei){
    E <- beta * E_rm + (n*E_ei)/n -0.01
    Sigma <- sqrt((beta^2 * sigma_M^2) + (n* sigma_i^2)*1.0/(n^2))

    return (E*1.0/Sigma)
}
```

- If $\mathbb{E}[\epsilon_i] = 1.5\sigma_i$

In [6]:
```
# Compute the Sharpe ratio w.r.t. N
N <- 2:30
SharpeRatio_res1 <- SharpeRatio(N,E_ei1)
SharpeRatio_res1
```

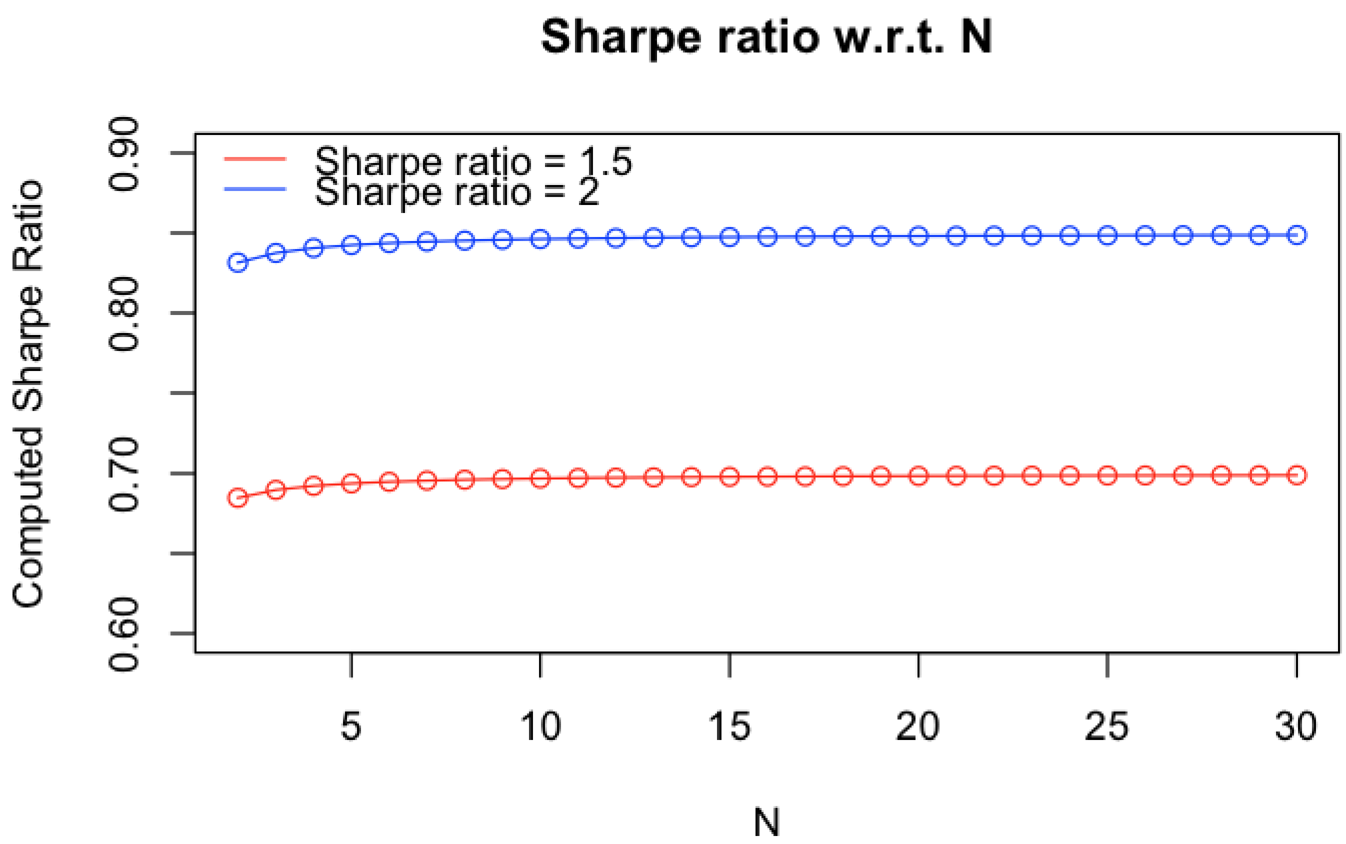
0.684762383262326 0.689730494715005 0.692255447007808 0.693783794022838 0.694808333779651
0.695542933104244 0.69609541422908 0.696526033146992 0.696871104276978 0.697153816990652
0.697389673941405 0.697589432290069 0.697760790423871 0.69790940299673 0.698039516921919
0.698154383781083 0.698256535275343 0.698347971998746 0.698430295769578 0.698504804273508
0.698572559973702 0.698634441096847 0.69869117990833 0.698743391826045 0.698791597832997
0.698836241921708 0.698877704809514 0.698916314822792 0.698952356609212

- If $\mathbb{E}[\epsilon_i] = 2\sigma_i$

In [7]:
```
SharpeRatio_res2 <- SharpeRatio(N,E_ei2)
SharpeRatio_res2
```

0.831497179675681 0.837529886439649 0.840595899938053 0.842451749884874 0.843695833875291
0.844587847340868 0.845258717278168 0.845781611678491 0.846200626622044 0.846543920631505
0.84683031835742 0.847072882066512 0.847280959800415 0.847461417924601 0.847619413405188
0.847758894591315 0.847882935691488 0.847993965998478 0.848093930577345 0.84818440518926
0.848266679968067 0.848341821331886 0.848410718460114 0.848474118645912 0.848532654511496
0.848586865190645 0.848637212982982 0.848684096570534 0.8487278615969

In [8]:
```
# Plot the two Sharpe ratio
options(repr.plot.height=4,repr.plot.width=6)
yrange <- c(0.6,0.9)
plot(N,SharpeRatio_res1,col="red",ylab="Computed Sharpe Ratio",xlab="N",
     main="Sharpe ratio w.r.t. N",type='o',ylim=yrange)
lines(N,SharpeRatio_res2,type='o',col="blue")
legend('topleft', c("Sharpe ratio = 1.5","Sharpe ratio = 2") , lty=1,bty='n',
       y.intersp=1.5,cex=1,col=c("red", "blue") )
```




### Answer of (d):

- If $\beta = 0, \sigma_i = 0.1$, we can compute the new computed Sharpe ratio as:

In [9]:
```
# Define a funciton  to compute the new Sharpe ratio
# Define a function to compute the sharpe ratio
beta <- 0
sigma_M <- 0.2
sigma_i <- 0.1
E_rm <- 0.07
E_ei1 <- 1.5*sigma_i
E_ei2 <- 2.0*sigma_i

SharpeRatio_new <- function(n,E_ei){
    E <- beta * E_rm + (n*E_ei)/n -0.01
    Sigma <- sqrt((beta^2 * sigma_M^2) + (n* sigma_i^2)*1.0/(n^2))

    return (E*1.0/Sigma)
}
```

In [10]:
```
# Compute the new Sharpe ratio
N <- 2:30
SharpeRatio_new_res1 <- SharpeRatio_new(N,E_ei1)
SharpeRatio_new_res2 <- SharpeRatio_new(N,E_ei2)
```

In [11]:
```
SharpeRatio_new_res1
```

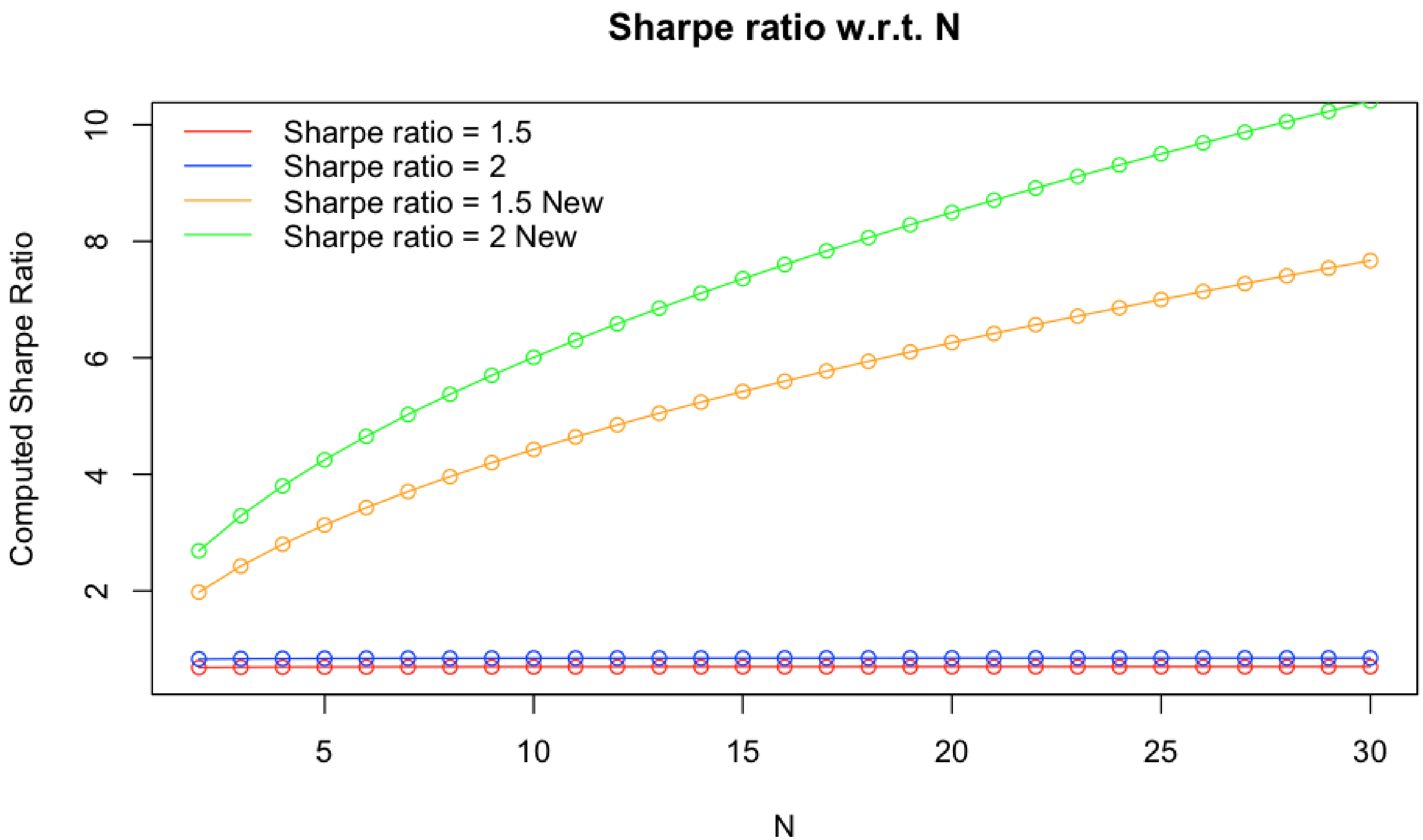
1.97989898732233 2.42487113059643 2.8 3.13049516849971 3.42928563989645 3.70405183549043
3.95979797464467 4.2 4.42718872423573 4.64327470649756 4.84974226119286 5.04777178564959
5.23832034148352 5.42217668469038 5.6 5.77234787586472 5.939696961967 6.10245852095694
6.26099033699941 6.41560597293818 6.5665820637528 6.71416413263781 6.8585712797929 7
7.1386273190299 7.27461339178928 7.40810367098085 7.53923072998831 7.66811580507233

In [12]:
```
SharpeRatio_new_res2
```

2.68700576850888 3.29089653438087 3.8 4.2485291572496 4.65403051128804 5.02692749102272
5.37401153701776 5.7 6.00832755431992 6.30158710167526 6.58179306876173 6.85054742338158
7.10914903487049 7.35866835779409 7.6 7.83390068867355 8.06101730552664 8.28190799272728
8.4970583144992 8.7068938204161 8.91178994366452 9.11207989429417 9.30806102257608 9.5
9.68813707582629 9.8726896031426 10.0538549820454 10.2318131335556 10.4067285925982

In [13]:

```
# Plot the comparison
options(repr.plot.height=5,repr.plot.width=8)
yrange <- c(0.6,10)
plot(N,SharpeRatio_res1,col="red",ylab="Computed Sharpe Ratio",xlab="N",
     main="Sharpe ratio w.r.t. N",type='o',ylim=yrange)
lines(N,SharpeRatio_res2,type='o',col="blue")
lines(N,SharpeRatio_new_res1,type='o',col="orange")
lines(N,SharpeRatio_new_res2,type='o',col="green")
legend('topleft', c("Sharpe ratio = 1.5","Sharpe ratio = 2","Sharpe ratio = 1.5 New","Sharpe ratio =
 2 New") ,
       lty=1,bty='n',
       y.intersp=1.5,cex=1,col=c("red", "blue","orange","green") )
```




- From the figure above, we can conclude that the investment method in (d) is better than (c).