SENIOR PROJECT FALL 2022

THE STUDY OF STOCK PRICES AND ANALYSIS OF FINANCIAL PORTFOLIO

Hasan Khan

UNIVERSITY OF HOUSTON-DOWNTOWN

Table of Contents

CHAPTER 1: Interests

- **Simple Interest:**

$\boldsymbol{A = P + Prt \ = \ P(1 + rt)}$

A: Accumulated value
P: Principal
r: nominal interest rate
t: duration in years

EXAMPLE

Given :
P = 1000
r = 5% = .05
t = 18 months = 18/12 = 1.5 years

$$A = \ 1000\big(1 + (.05)(1.5)\big)$$
$$A = \$1075$$

- **Compound Interest:**

$\boldsymbol{A = P(1 + r)^t}$

A: Accumulated value
P: Principal
r: nominal interest rate
t: duration in years

EXAMPLE

P = 1000
r = 5% = .05
t = 18 months = 18/12 = 1.5 years

$$A = 1000(1 + .05)^{1.5}$$
$$A = 1075.93$$

t	A
0	P

1	$A_1 = \mathrm{P}(1+\mathrm{r})^1$
2	$A_2 = \mathrm{P}(1+\mathrm{r})^2 = A_1(1+r)$
3	$A_3 = A_2(1+r)$

- **Periodic Compounding**:

$$A = P(1+\frac{r^{(m)}}{m})^{m.t} = P(1+j_{(m)})^n$$

m : Compound frequency
$\boldsymbol{r^{(m)}}$**:** Nominal yearly rate
$\boldsymbol{j_{(m)} = \frac{r^{(m)}}{m}}$ **:** Rate-per-period

n=m.t : number of periods

EXAMPLE:

P = 1000
r = 5%
m = 12(monthly)
t = 3 years

$$A = 1000\left(1+\frac{.05}{12}\right)^{12(3)}$$
$$A = 1161.49$$

Quarterly:

$$A = 1000\left(1+\frac{.05}{4}\right)^{4(3)}$$
$$A = \$1160.75$$
Note: Monthly gives more interest as compared to quarterly.

- **EQUIVALENT RATES**

Two nominal rates $r^{(m)}, r^{(p)}$ are 'equivalent' if the generate the same accumulated values, over 1 year

$$\cancel{P}(1+\frac{r^m}{m})^{m.t} = \cancel{P}(1+\frac{r^p}{p})^{p.t}$$

$$\left(1+\frac{r^{(m)}}{m}\right)^m = \left(1+\frac{r^{(p)}}{p}\right)^p$$

- **Effective Annualized Rate (EAR)**:

  EAR is defined as

  $$1 + EAR = \left(1 + \frac{r^{(m)}}{m}\right)^m$$

EXAMPLE:

(a) Given $r^{(12)} = .05$; $EAR = \left(1 + \frac{.05}{12}\right)^{12} - 1 = .05116$

(b) Given EAR = .05; $1 + .05 = \left(1 + \frac{r^{(12)}}{12}\right)^{12}$; $\frac{r^{(12)}}{12} = (1 + .05)^{\frac{1}{12}} - 1 =$ $.00407$; $r^{(12)} = 12(.00407) =$ $.04889$

Note: EAR Is larger or equal to nominal rates.

- **Continuous Compounding**

  $\boldsymbol{A = Pe^{rt}}$

  r: nominal rate

  EXAMPLE:

  $P = 1000$;
  $r = .05$
  $t = 5\ yrs$
  $A = 1000e^{(.05)(5)} = 1284.03$

CHAPTER 2: ANNUITIES

- **Annuity**: a string of payments (mostly fixed)
- **Annuity – Immediate:** n payments, each made at the end of period
- **Annuity-Due**: n payments, each made at the beginning of period

PV:Present value
FV: Future value
CF: Cash flow

- **Annuity-Immediate Present Value**

$PV = CFa_{n,i}$, where

$$a_{n,i} = \frac{1 - v^n}{i} = \frac{1 - (1 + i)^{-n}}{i}$$

- **Annuity-Immediate Future Value**

$FV = CVs_{n,i}$, where

$$s_{n,i} = \frac{(1 + i)^n - 1}{i}$$

- **Annuity-Due Present Value**

$PV = CF\,\ddot{a}_{n,i}$, where

$$\ddot{a}_{n,i} = \frac{1 - v^n}{d} = \frac{1 - (1 + i)^{-n}}{d}$$

- **Annuity-Due Future Value**

$FV = CF\ddot{s}_{n,i}$

$$\ddot{s}_{n,i} = \frac{(1 + i)^n - 1}{d}$$

$$i = \frac{j_{(m)}}{m} = rate - per - period$$
$$n = mt = number\ of\ periods$$
$$v^n = (1 + i)^{-n}$$

$$d = \frac{i}{1+i}$$

EXAMPLE:
(a) Retirement payments of $1,000 is paid to you for 10 years with a rate-per-period of 5% and compounded monthly. Find the annuity-immediate present value and its future value.

Answer: The number of periods is $n = 10 \times 12 = 120$ and the yield rate is $i = j_{(12)} = \frac{r^{(12)}}{12} = \frac{.05}{12} = .0042$.

The annuity-immediate PV is $a_{120\neg .0042} = \$1{,}000\left[\frac{1-(1+.0042)^{-120}}{.0042}\right] = \$94{,}107.78$.

You must have this amount at this current time for those payments.

(b)Suppose payments of $1,000 are paid into a retirement accound for 10 years, with a rate-per-period of 5% and compounded monthly. Find the annuity-immediate future value.

ANS: The annuity-immediate future value is FV = CF* $s_{120\neg .0042} = \$1{,}000 \times [\frac{(1+.0024)^{120}-1}{.0024}] = \$138{,}873.82$.

You will have $138,873.82 In 10 years.

NOTE: As interest rate is increasing, the present value decreases.

CHAPTER 3: STOCK PRICES AND RETURNS

3.1 Returns and Log-returns

- **Returns**
  P(t) : price at time t t:discrete=0,1,2,3,...

  Return: $R(t) = \frac{P(t)-P(t-1)}{P(t-1)} = \frac{P(t)}{P(t-1)} - 1$

  $\frac{P(t)}{P(t-1)} = 1 + R(t)$ (1)

  $P(t) = P(t-1)\big(1 + R(t)\big) = \cdots =$
  $P(t) = P(0)\big(1 + R(1)\big)\big(1 + R(2)\big) \ldots (1 + R(t))$

  Another notation (2)
  $P(t) = P(o)\,\Pi_{i=1}^{t}\big(1 + R(i)\big)$

- **Log-returns**

From (1)

$\ln\big(P(t)\big) - \ln\big(P(t-1)\big) = \ln\big(1 + R(t)\big)$

Define log-return: $r(t) = \ln\,(1 + R(t))$
$\ln\big(P(t)\big) - \ln\big(P(t-1)\big) = r(t)$

From (2)

$\ln\big(P(t)\big) = lnP(0) + \Sigma_{i=1}^{t}\, r(i)$

NOTE:
(i) Returns and log-returns are used Interchangeable

(ii)If R(t) Is small (≈ 0) then R(t) $\approx r(t)$

3.2 Weakly stationary

Weakly stationary
$Y(t), t = 1,2,3, \ldots$ Is a time-series

(1) Mean function: $\mu(t) = E[Y(t)] = \frac{\Sigma^t Y(i)}{t}$
(2) Variance function: $\sigma^2(t) = Var[Y(t)] = \frac{\Sigma^t \left(Y(i) - \mu(t)\right)^2}{t-1}$

(3) Auto-Covariance :$\gamma(t, s) = cov\big(Y(t), Y(s)\big)$

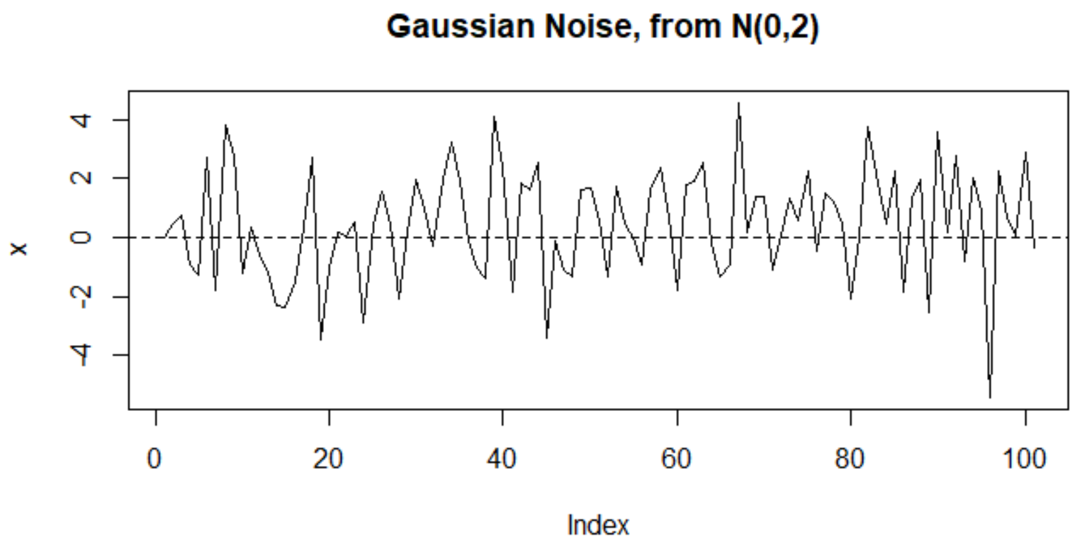
(4) Auto-Correlation: $\rho(t, s) = \frac{cov(t,s)}{\sqrt{Var(Y_t)} * \sqrt{Var(Y_s)}}$

Time series $\{ Y(t), t = 1,2, .. \}$ is **weakly stationary** if
(1) $E[Y(t)] = \mu$
(2) $Var[Y(t)] = \sigma^2$
(3) $cov(Y_t, Y_s) = \gamma(|t - s|), \quad any\ s, t$
(4) $cor(Y_t, Y_s) = \rho(|t - s|), \quad any\ s, t$

Note: Most Financial time series fail to be weakly stationary. Other models must be explored.

3.3 : NOISE,RANDOM WALK,WIENER PROCESS, GENERALIZED WIENER PROCESS.

Noise: $Z(t)$ is called a Noise if:
(1) $E[\,Z(t)] = 0$
(2) $Var[Z(t)] = \sigma^2$
(3) $cov\,\big(Z(s), Z(t)\big) = 0$ [uncorrelated]

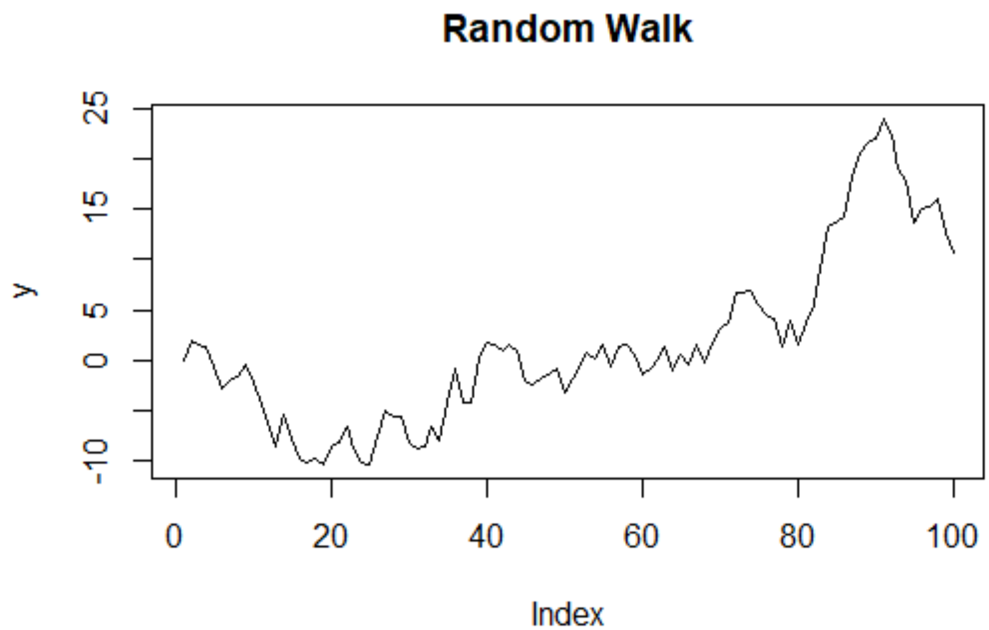



```
x <- c(0, rnorm(99, 0,2))   # this Noise starts at 0.  x(0)=0
plot(x, type="l",  main="Gaussian noise, N(0,2)")
```

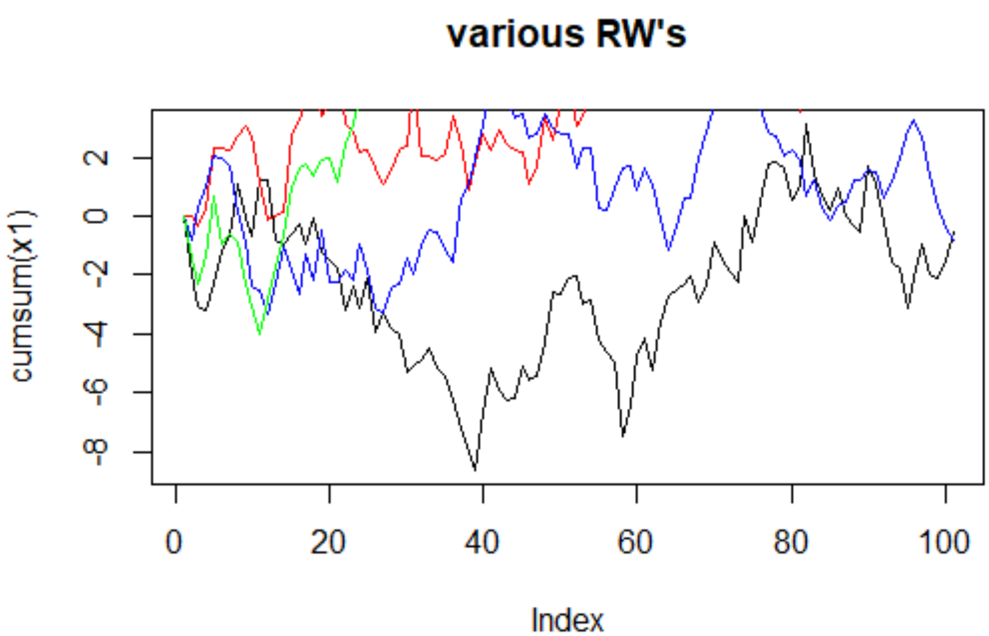
REMARK: Noise Is Weakly stationary, by definition

Random Walk:
Y(t) is a Random Walk if:
$Y(t) = \Sigma Z(t)$, where Z(t) is a noise

If Y(t) is a random walk then,
(1) $E[Y(t)] = 0$
(2) $Var[Y(t)] = t\sigma^2$
(3) $cov\left(Y(t), Y(s)\right) = 0$



REMARK: (1) Graph eventual returns to 0. (2) Graph gets wider and wider




```
y<- cumsum(x)
plot(y, type="l", main="Random Walk")
```

Wiener Process (or Brownian Motion):

W(t) is a Wiener process(or Brownian motion) if the increments, $\Delta w(t) = w(t+\Delta t) - w(t)$ satisfy

$\Delta w(t) = \epsilon\sqrt{\Delta t}$ where $\epsilon \sim N(0,1)$

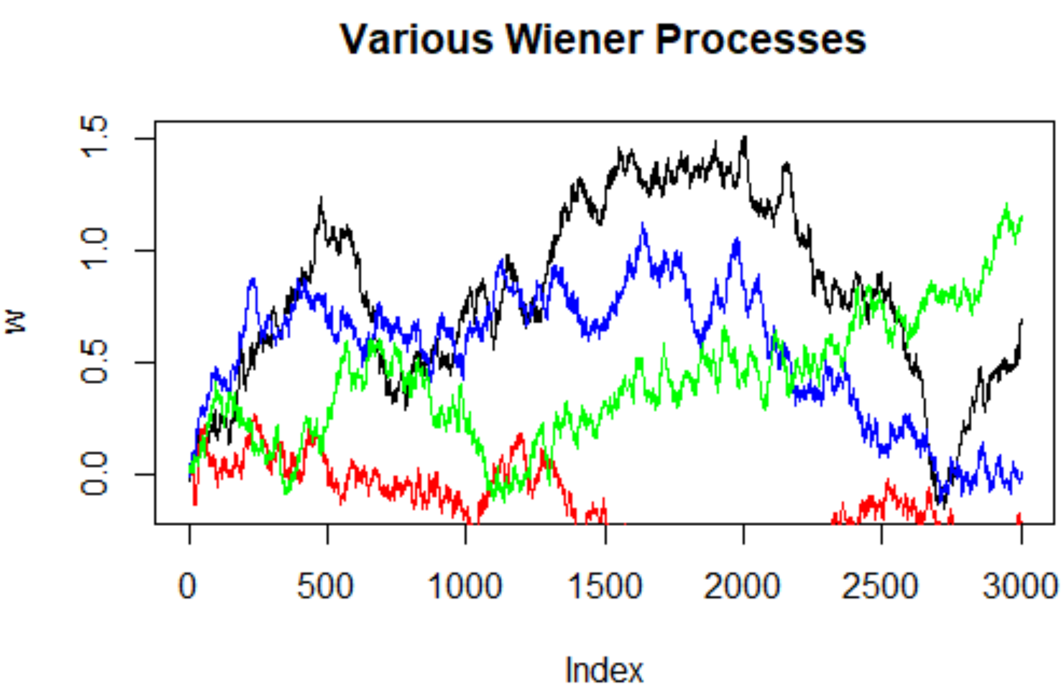
Properties:

(1) $\Delta w(t) = \epsilon\sqrt{\Delta t}$ where $\epsilon \sim N(0,1)$. Equivalently,

$\Delta w(t) = N\ (0, \sqrt{\Delta t}\ )$.

(2) Since w(t)=w(0)+ $\Delta w(1)$+ $\Delta w(2)$+… $\Delta w(t) = w(0) + \Sigma\Delta w(i)$, w(t) Is a Random Walk.

(2a) E[w(t)]=w(0)
(2b) Var[w(t)]=t(Δt)



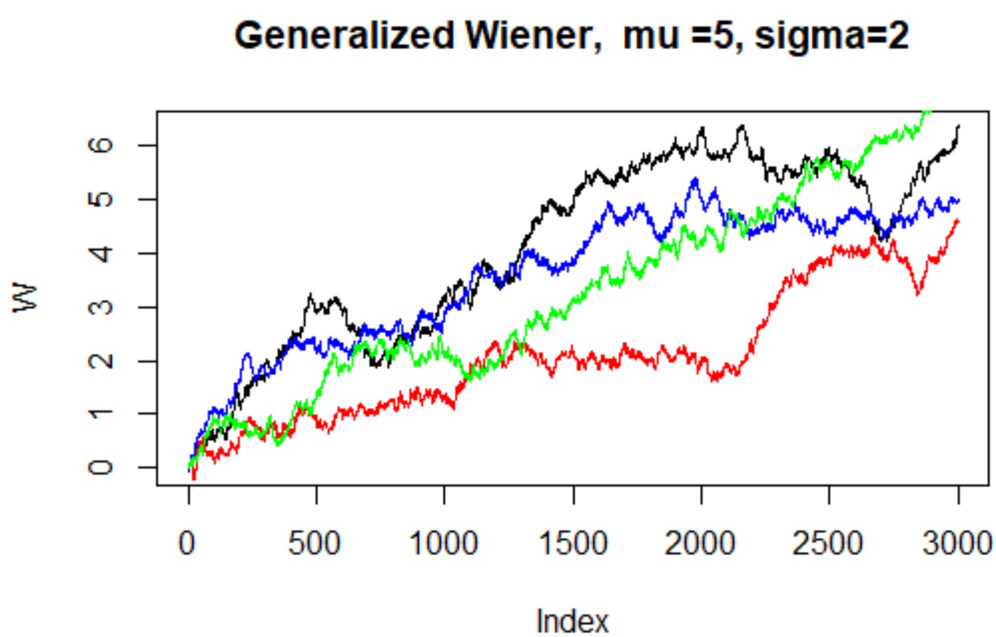
Wiener process with suppose t=1 (unit); n=3000

```
epsilon<- rnorm(n)   # epsilon
w <- cumsum(epsilon*sqrt(1/n))
plot(w, type="l", main="Various Wiener Processes")
w2<-cumsum(rnorm(n)*sqrt(1/n))
w3<-cumsum(rnorm(n)*sqrt(1/n))
w4<-cumsum(rnorm(n)*sqrt(1/n))
lines(w2, col="red")
lines(w3, col="blue")
lines(w4, col="green")
```

Generalized Wiener Process

$dX(t) = \mu dt + \sigma dW(t),$

Where constants
$\mu: Drift\,;\quad \sigma: Volatility$
$W(t)$ *is a wiener process*



REMARK: Graph would not return to zero.

- **Geometric Wiener Process (or Geometric Brownian Motion)**

$$dX(t) = \mu X dt + \sigma X \epsilon \sqrt{dt}$$

3.4 Properties of Price P(t) , Log Price p(t); Return R(t), Log-return r(t)

(1) Price P(t)

Price P(t):

$$\frac{P(t)}{P(t-1)} = 1 + R(t) \qquad (*)$$

By Recursiveness:

$$P(t) = P(0)\ \Pi_{i=1}^{t}(1 + R(i))$$

Note: P(t) Is a product of $(1 + R(i))$

(2) Log-price p(t)

Take log of (*)

$$ln\big(P(t)\big) - ln\big(P(t-1)\big) = ln\left(\frac{P(t)}{P(t-1)}\right) = ln(1 + R(t)) = r(t)$$

Log-price p(t):

$p(t) - p(t-1) = r(t)$

By recursiveness:

$$p(t) - p(0) = \Sigma^{t}{}_{i=1} r(i)$$

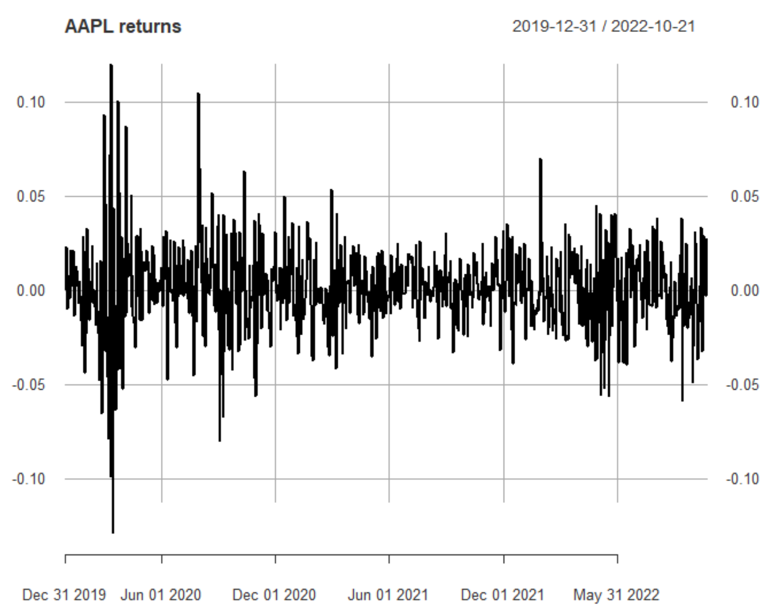
NOTE: (1) If $r(i)$'s are distributed Normal then the log price $p(t)$ is a sum of Noise , hence a Random Walk

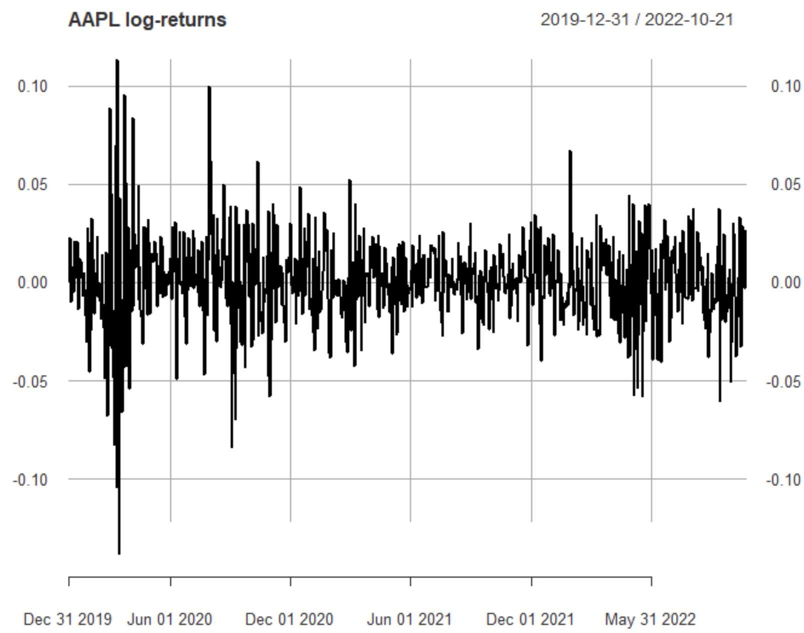
(2) $P(t) = P(0)e^{\Sigma^{t}{}_{i=1} r(i)}$

(3) We can check the assumptions of the Random walk hypothesis using the Box-Ljung test.

```
start<- as.Date("2019-12-31")
end<- as.Date("2022-10-22")

AAPL <-getSymbols("AAPL",from=start, to=end, auto.assign=F)

> head(AAPL)
        AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume
AAPL.Adjusted
2019-12-31  72.4825   73.420  72.3800   73.4125   100805600
72.03988
2020-01-02  74.0600   75.150  73.7975   75.0875   135480400
73.68357
2020-01-03  74.2875   75.145  74.1250   74.3575   146322800
72.96722
2020-01-06  73.4475   74.990  73.1875   74.9500   118387200
73.54863
2020-01-07  74.9600   75.225  74.3700   74.5975   108872000
73.20273
2020-01-08  74.2900   76.110  74.2900   75.7975   132079200
74.38029


> return.AAPL <- dailyReturn(AAPL$AAPL.Close)
> head(return.AAPL)
        daily.returns
2019-12-31  0.000000000
2020-01-02  0.022816333
2020-01-03 -0.009722044
2020-01-06  0.007968248
2020-01-07 -0.004703042
2020-01-08  0.016086289

plot(AAPL.Close, type="l", main="AAPL Close")
```

```
> plot(return.AAPL, type="l", main="AAPL returns")
> plot(log(1+return.AAPL), type="l", main="AAPL log-returns")
>
```





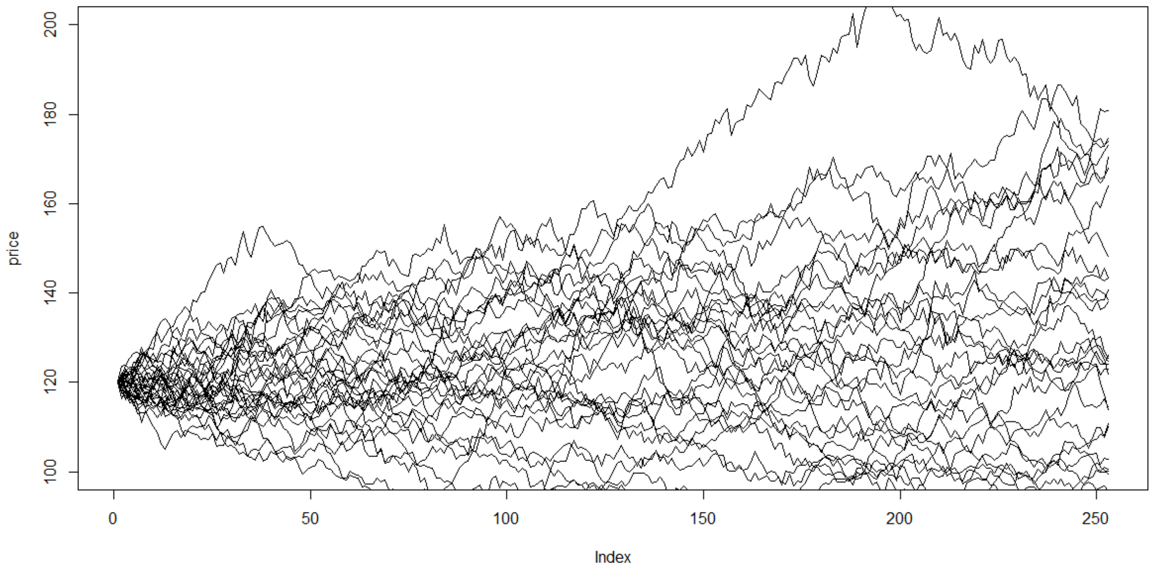

```
> Box.test(logreturn.AAPL, lag=365)

        Box-Pierce test

data:  logreturn.AAPL
X-squared = 334.24, df = 365, p-value = 0.8743
```

CHAPTER 4: STOCK PRICE SIMULATION

```
######################################
#### (1) A function to calculate mu, sigma
######################################


mu.sigma<- function(sample, lag=1){

  N<-length(sample)
  if (N < 1+lag){

    stop("sample must be greater than 2 +lag")
  }
```

```
 ct <- sample[(1+lag):N]
 pt<- sample[1: (N-lag)]
 t=1
 dt=t/N
 returns <- (ct-pt)/pt

 logreturns <- log(1+returns)
 logreturns.bar  <- mean(logreturns)

 s <- sd(logreturns)


 drift <- logreturns.bar*N  + s^2*N/2

 volatility <- sqrt(s^2*N)

 #cat("mu =",    round(drift, 4) ,"sigma=",round(volatility,4) ,   "\n")

 c(drift, volatility)

}
```

```
############################################################
########(2) Simulating a geometric random walk , in a span ########
of 1 year
############################################################

n = 252

logr = rnorm(n,0.05/253,0.2/sqrt(252))
price = c(120,120*exp(cumsum(logr)))
plot(price, type="l", ylim=c(100, 200))
for (i in (1:30))
{
 logr = rnorm(n,0.05/253,0.2/sqrt(253))
 price = c(120,120*exp(cumsum(logr)))
 lines(price)
}
```



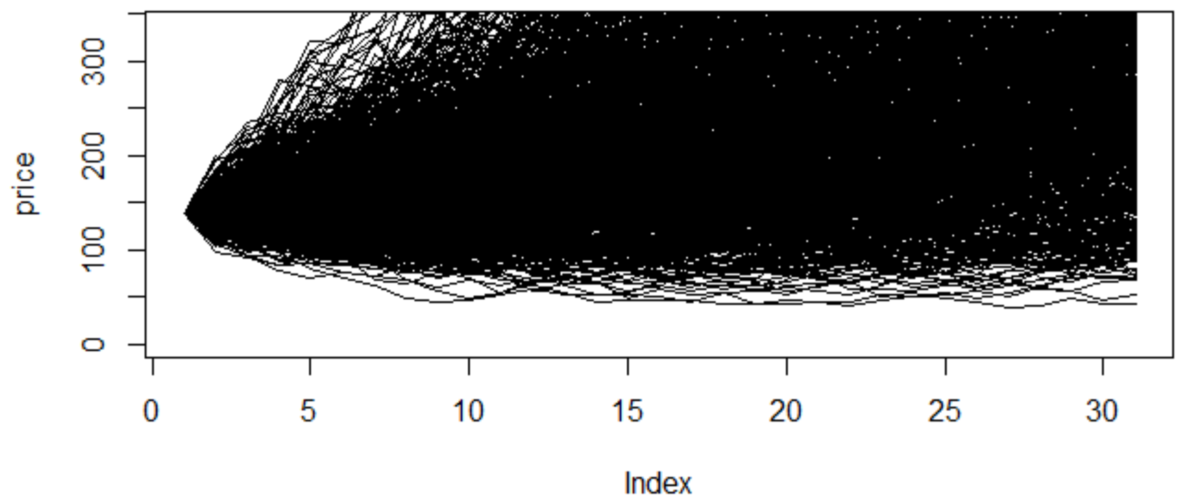

```
##################################################
##(3)  A function to simulate prices
##################################################
asset.price.sim <- function(mu, sigma, initprice, N, iter){


 logr = rnorm(N,mu/N,sigma/sqrt(N))
 price = c(initprice,initprice*exp(cumsum(logr)))
 plot(price, type="l", ylim=c(0, initprice+200))

 price.mat<- rep(0, N)
 for (i in (1:iter))
 {
  logr = rnorm(N,mu/N,sigma/sqrt(N))
   price = c(initprice,initprice*exp(cumsum(logr)))
```

```
      lines(price)

      price.mat <- cbind(price.mat, price)

  }

  means<- rowMeans(price.mat)

  c(means)

}
```

```
##################################################
### (4) EXAMPLE: AAPL Close Price; Simulate 30 prices
###################################################
```

Historical data from="2019-12-31", to="2022-10-01 from AAPL Close Price

```
> AAPL.Prices <- getSymbols("AAPL", from="2019-12-31", to="2022-10-01",
auto.assign=FALSE)
> AAPL.Close <- Cl(AAPL.Prices)
> data<- as.vector(AAPL.Close$AAPL.Close)
> mu.sigma(data, lag=1)
[1] 0.8184459 0.6081535

result<- c(0.8184459, 0.6081535)
price.sim<-asset.price.sim(mu=result[1] , sigma=result[2],
initprice=73.4125, N=30, iter=1000)
```

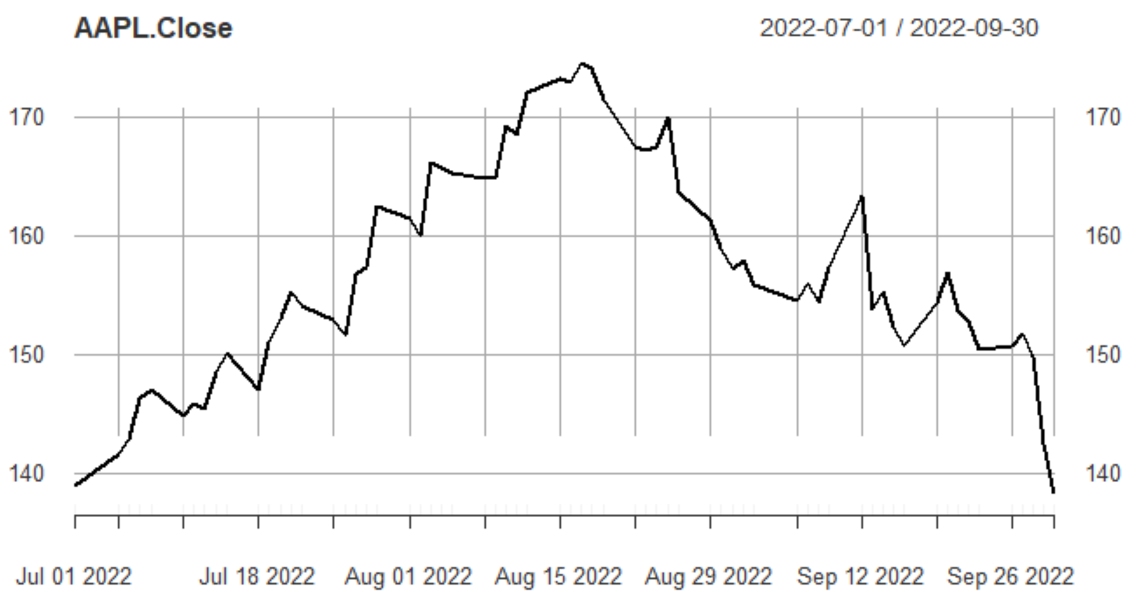


##
(5) AAPL Close prices(from from="2022-07-01",
to="2022-10-01")
###

Train data set

```
AAPL.Prices <- getSymbols("AAPL", from="2022-07-01", to="2022-10-01",
auto.assign=FALSE)
> AAPL.Close <- Cl(AAPL.Prices)
> head(AAPL.Close)
           AAPL.Close
2022-07-01     138.93
2022-07-05     141.56
2022-07-06     142.92
2022-07-07     146.35
2022-07-08     147.04
2022-07-11     144.87
> tail(AAPL.Close)
           AAPL.Close
2022-09-23     150.43
2022-09-26     150.77
2022-09-27     151.76
```

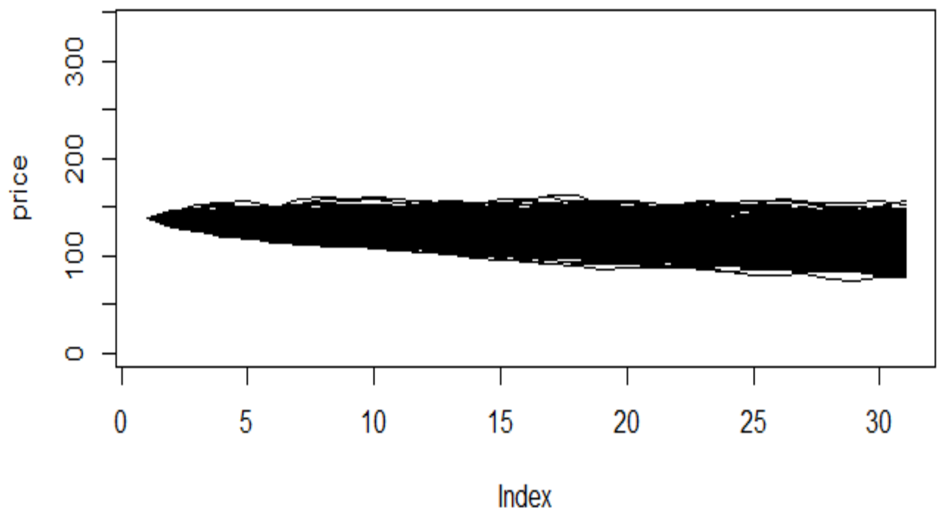
2022-09-28   149.84
2022-09-29   142.48
2022-09-30   138.20



train<- as.vector(AAPL.Close$AAPL.Close)

Mu-Sigma of AAPL.Close
> result<-mu.sigma(train, lag=1)
> result
[1] 0.006668355 0.155049978
>

Simulation of prices
> price.sim<- asset.price.sim(mu=result[1], sigma=result[2], **138.20**, 30, 1000)
$rowMeans.price.mat.
 [1] 138.0619 138.1258 138.0171 137.9497 137.9343 137.8880 138.0925 138.3127
 [9] 138.3229 138.5447 138.2764 138.4823 138.5577 138.7653 138.9668 139.0225
[17] 139.2698 139.0708 138.8477 138.9165 139.0999 139.2155 139.4848 139.6538
[25] 139.8181 139.9403 140.1219 140.1249 140.2621 140.2316 140.3753

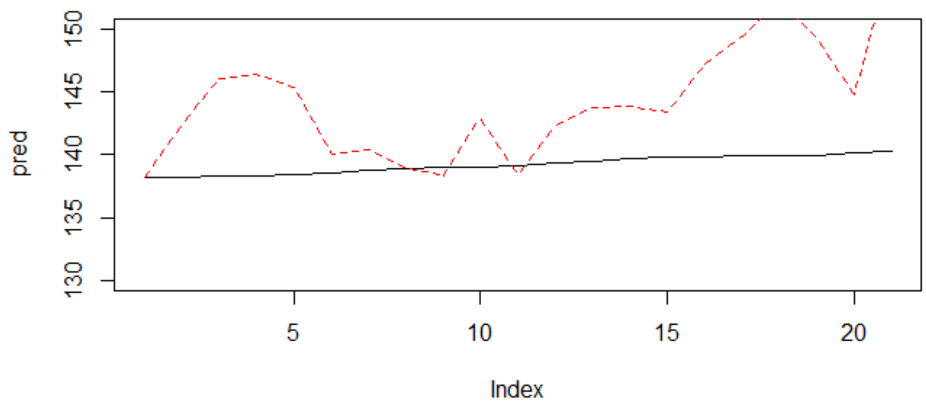



```
pred<- data.frame(price.sim[1:21])

rownames(pred)<-index(AAPL.Close.test)

> pred
          price.sim.1.21.
2022-09-30       138.0619
2022-10-03       138.4518
2022-10-04       138.1990
2022-10-05       138.3066
2022-10-06       138.4451
2022-10-07       138.7966
2022-10-10       138.9866
2022-10-11       139.0302
2022-10-12       139.3595
2022-10-13       139.5967
2022-10-14       139.8388
2022-10-17       139.8426
2022-10-18       139.7793
2022-10-19       139.6344
2022-10-20       139.6142
2022-10-21       139.4804
2022-10-24       139.6005
2022-10-25       139.7376
2022-10-26       139.7527
```

```
2022-10-27      140.0167
2022-10-28      140.0024
```

> **## Test data set (21 days)**

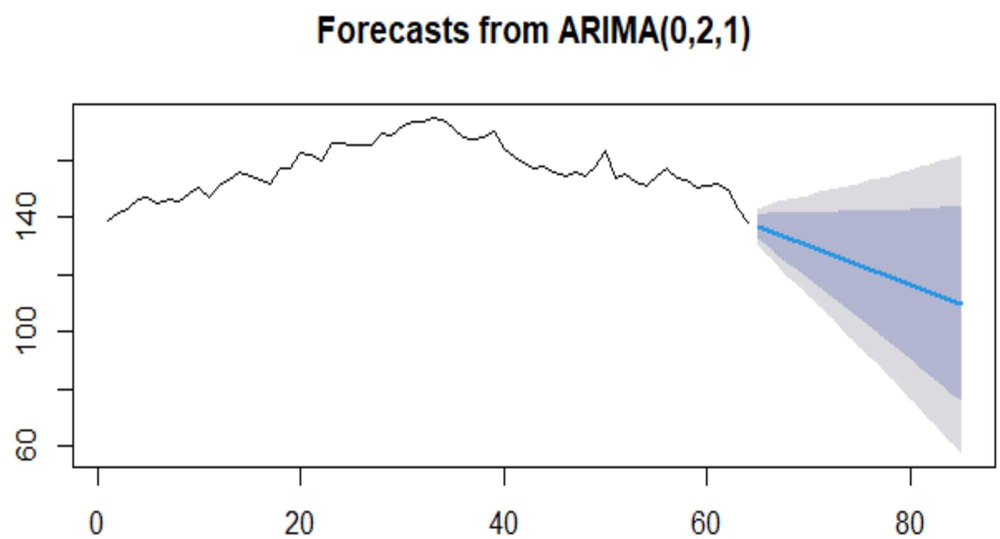
```
> AAPL.test <- getSymbols("AAPL", from="2022-09-30", to="2022-10-30",
auto.assign=F)
> AAPL.Close.test <- Cl(AAPL.test)
> AAPL.Close.test
           AAPL.Close
2022-09-30    138.20
2022-10-03    142.45
2022-10-04    146.10
2022-10-05    146.40
2022-10-06    145.43
2022-10-07    140.09
2022-10-10    140.42
2022-10-11    138.98
2022-10-12    138.34
2022-10-13    142.99
2022-10-14    138.38
2022-10-17    142.41
2022-10-18    143.75
2022-10-19    143.86
2022-10-20    143.39
2022-10-21    147.27
2022-10-24    149.45
2022-10-25    152.34
2022-10-26    149.35
2022-10-27    144.80
2022-10-28    155.74
```




```
##################################################
##### (6) Use of Time Series Models (ARIMA)
##################################################

> library(forecast)
Warning message:
package 'forecast' was built under R version 4.1.3
> model <- auto.arima(train)
> summary(model)
Series: train
ARIMA(0,2,1)

Coefficients:
          ma1
      -0.9154
s.e.   0.0493

sigma^2 = 9.168:  log likelihood = -157.06
AIC=318.13   AICc=318.33   BIC=322.38

Training set error measures:
                   ME     RMSE      MAE        MPE     MAPE      MASE
Training set -0.5348308 2.955982 2.305122 -0.3587578 1.470804
0.9948801
                   ACF1
Training set -0.1427278


> forecast <- forecast(model, h=21)
```

```
> summary(forecast)

Forecast method: ARIMA(0,2,1)

Model Information:
Series: train
ARIMA(0,2,1)

Coefficients:
        ma1
    -0.9154
s.e.  0.0493

sigma^2 = 9.168:  log likelihood = -157.06
AIC=318.13   AICc=318.33   BIC=322.38

Error measures:
                 ME     RMSE      MAE       MPE     MAPE      MASE
Training set -0.5348308 2.955982 2.305122 -0.3587578 1.470804
0.9948801
               ACF1
Training set -0.1427278

Forecasts:
  Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
65       136.8470 132.96668 140.7272 130.91258 142.7813
66       135.4939 129.76955 141.2183 126.73924 144.2486
67       134.1409 126.83669 141.4451 122.97007 145.3117
68       132.7879 124.01172 141.5640 119.36590 146.2098
69       131.4348 121.23718 141.6325 115.83888 147.0308
70       130.0818 118.48522 141.6784 112.34636 147.8172
71       128.7288 115.74022 141.7173 108.86449 148.5930
72       127.3757 112.99267 141.7588 105.37873 149.3727
73       126.0227 110.23641 141.8090 101.87966 150.1657
74       124.6697 107.46732 141.8720  98.36096 150.9784
75       123.3166 104.68254 141.9507  94.81825 151.8150
76       121.9636 101.88006 142.0471  91.24847 152.6787
77       120.6106  99.05844 142.1627  87.64943 153.5717
78       119.2575  96.21667 142.2984  84.01958 154.4955
79       117.9045  93.35403 142.4550  80.35780 155.4512
80       116.5515  90.47001 142.6329  76.66332 156.4396
81       115.1984  87.56426 142.8326  72.93561 157.4612
82       113.8454  84.63656 143.0542  69.17434 158.5165
83       112.4924  81.68678 143.2979  65.37928 159.6054
84       111.1393  78.71485 143.5638  61.55037 160.7283
85       109.7863  75.72077 143.8518  57.68757 161.8850
```

```
> plot(forecast)
>
```




```
##################################
#### (6) Compare 2 models: MSE and RMSE
##################################

> library(ModelMetrics)
>forecast<- data.frame(forecast)

 > rmse(pred$price.sim.1.21., AAPL.Close.test$AAPL.Close)
 [1] 6.707402
 > rmse(forecast$Point.Forecast, AAPL.Close.test$AAPL.Close)
 [1] 23.97453
```

REMARK: The Random Walk approach show a smaller MSE.

##
(7) Probability Distribution and Confidence Interval ##### for Price
##

For S(t) which are distributed as Log-Normal(μ,σ)

(1) $E[S(t)] = S(0)e^{\left(\mu+\frac{1}{2}\sigma^2\right)t}$

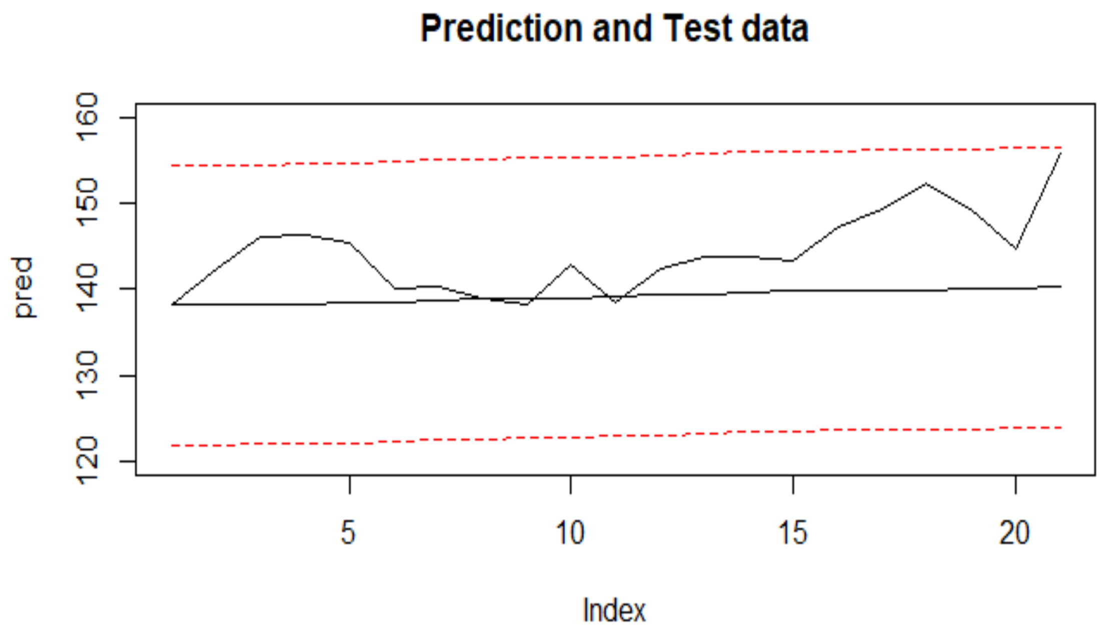
$E[S(t)] = S(0)e^{\mu+\frac{1}{2}\sigma^2}, for\ \ t = 1$

(2) $E[S(t)^2] = S(0)^2\ e^{(2\mu+2\sigma^2)t}$

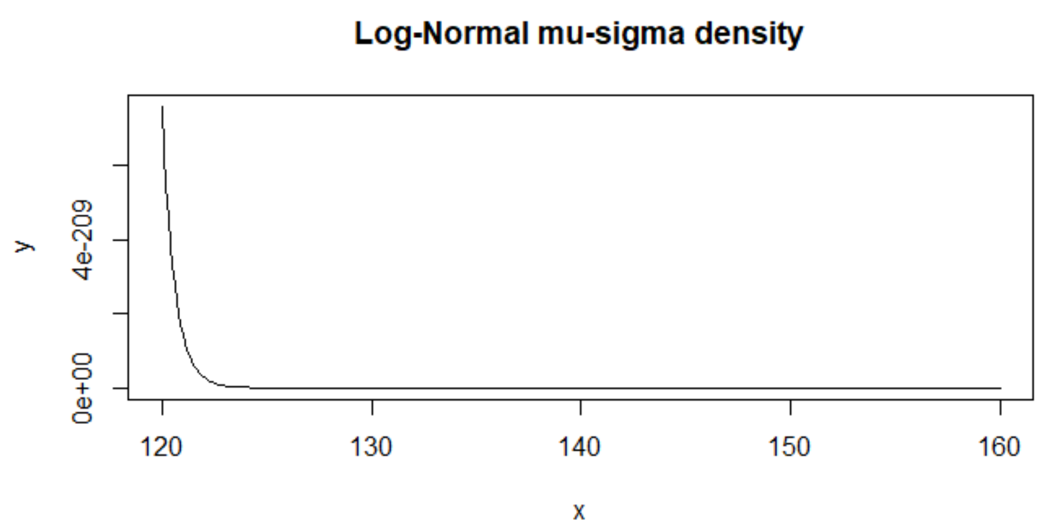
$E[S(t)^2] = S(0)^2\ e^{2\mu+2\sigma^2},\ for\ t = 1$

(3) $Var[S(t)] = E[S(t)^2] - (E[S(t)])^2$

```
> S0 <-138.2
> mu<-result[1]
> mu
[1] 0.006668355
> sigma<- result[2]
> E.price <- S0*exp(mu +(1/2)*sigma^2)
> E.price
[1] 140.8071
> E.price.sq <- S0^2*exp(2*mu+2*sigma^2)
> Var.price <- E.price.sq - E.price^2


> Sd.price <- sqrt(Var.price)
> Sd.price
[1] 21.964
> lower.critical<- qlnorm(.025, mu, sigma)
> upper.critical<- qlnorm(.975, mu, sigma)
> head(pred)
[1] 138.1724 138.1871 138.2580 138.3018 138.3712 138.5334
> LOWER<- pred -lower.critical*Sd.price
> UPPER<- pred +lower.critical*Sd.price

> plot(pred, type="l",  main="Prediction and Test data", ylim=c(120,160))
> lines(AAPL.real)
> lines(UPPER, lty="dashed", col="red")
> lines(LOWER, lty="dashed", col="red")
```




```
################################################
###   Log-Normal Distribution of Price
################################################
> x<- seq(120, 160, by=.1)
> y<- dlnorm(x, mu,sigma)
> plot(x,y, type="l", main="Log-Normal mu-sigma density")
```



```
## Probability of Price > 140$  is 0

> plnorm(140.8 , mu, sigma, lower.tail=FALSE)
[1] 4.075534e-223


## Probablity of Price < 140$ is 1
> plnorm(140.8 , mu, sigma, lower.tail=TRUE)
[1] 1
> plnorm(130.0 , mu, sigma, lower.tail=TRUE)
[1] 1
```

Price is decreasing in certainty

CHAPTER 4: FINANCIAL PORTFOLIO ANALYSIS

4.1 Portfolio

A Portfolio : a collection of various assets: bonds, stocks. Bonds are considered to be risk-free asset. Stocks are risky assets.

Value of a Portfolio, V(t) , Is the value of whole portfolio, whichh depends on the value of each asset , at a specific instant t (month, year).

Return of Portfolio: Suppose there are N assets: $i = 1,2,..,\ N$ assets (risk-free and risky), then at a specific instant t,

$$V_P(t) = V_1(t) + V_2(t) + \ ... + V_N(t)$$

Return of the Portfolio:

$$R(t) = \frac{V(t)-V(t-1)}{V(t-1)}$$

NOTE: $R(t)$ Is a random variable

4.2 Expected return /Variance of portfolio return.

Number of shares n_i ,
Stock price per share: $S_i(t)$
Weight of an asset : $w(i)$

- Value of i-th asset: $V_i(t) = n_i * S_i(t)$
- Weight of an asset: $w_i = \frac{V_i(t)}{V_P(t)}$

$$\boxed{w_i = \frac{V_i(t)}{V_P(t)} = \frac{n_i * S_i(t)}{V_P(t)}}$$

Given N assets in a portfolio.

weight of an asset: w_i

(1) $R(t) = \ w_1 R_1(t) \ + w_2 R_2(t) \ + \cdots + \ w_N R_N(t)$

(2) Expectation: $\mu_P = R_P(t) = w_1 \mu_{R_1}(t) + \cdots + w_n \mu_{R_N}(t)$

(3) Variance: $\sigma^2{}_P = Var(R_P) = w_1^2 \, \sigma^2{}_1 + .. + w_N^2 \sigma^2{}_2 + 2 \, \Sigma_i \Sigma_j \, w_i w_j Cov\big(R_i, \, R_j\big)$

$= w_1^2 \, \sigma^2{}_1 + .. + w_N^2 \sigma^2{}_2 + 2 \, \Sigma_i \Sigma_j \, w_i w_j \sigma_i \sigma_j \, \rho_{ij}$ (3)
Where $\rho_{ij} \ = \ cor(\, R_i, R_j \,)$

Risk: $\sigma_P \ = \sqrt{\sigma_P^2}$

PROOF of (1)

- $R_P(t) = \frac{V(t)}{V(t-1)} - 1 = \frac{\Sigma V_i(t)}{\Sigma \, V_i(t-1)} - 1 = \frac{\Sigma V_i(t) - \Sigma \, V_i(t-1)}{\Sigma V_i(t-1)} =$
$= \frac{V_1(t) - V_1(t-1) + V_2(t) - V_2(t-1) + .. + V_N(t) - V_N(t-1)}{\Sigma V_i(t-1)} =$
$= \frac{V_1(t) - V_1(t-1)}{V_1(t-1)} * \frac{V_1(t-1)}{\Sigma V_i(t-1)} + \cdots + \frac{V_N(t) - V_N(t-1)}{V_N(t-1)} * \frac{V_N(t-1)}{\Sigma V_i(t-1)}$
$= w_1 R_1(t) \ + \ \ w_2 R_2(t) + \cdots + w_N R_N(t)$ DONE

PROOF of (2) (3) follows from (1)

Remarks:
(a) Portfolio return at any given time is a weighted average of asset returns;
(b) Portfolio Expected return is We**ighted Average** of the asset expected returns

4.3 Mean-Variance plane . Weights constraint ; Efficient Frontier;

There are 2 variables of interest:

$\mu_R \ = E[R]$ Expected return	
$\sigma_R \ = Var[R]$ Risk	

The Optimization Problem:

Given the **constraint**

$$w_1 \ + w_2 \ + \cdots \ + w_N \ \ = 1 \ (*)$$

We need to solve the Optimization problem **objective function (Portfolio Return or Risk),**

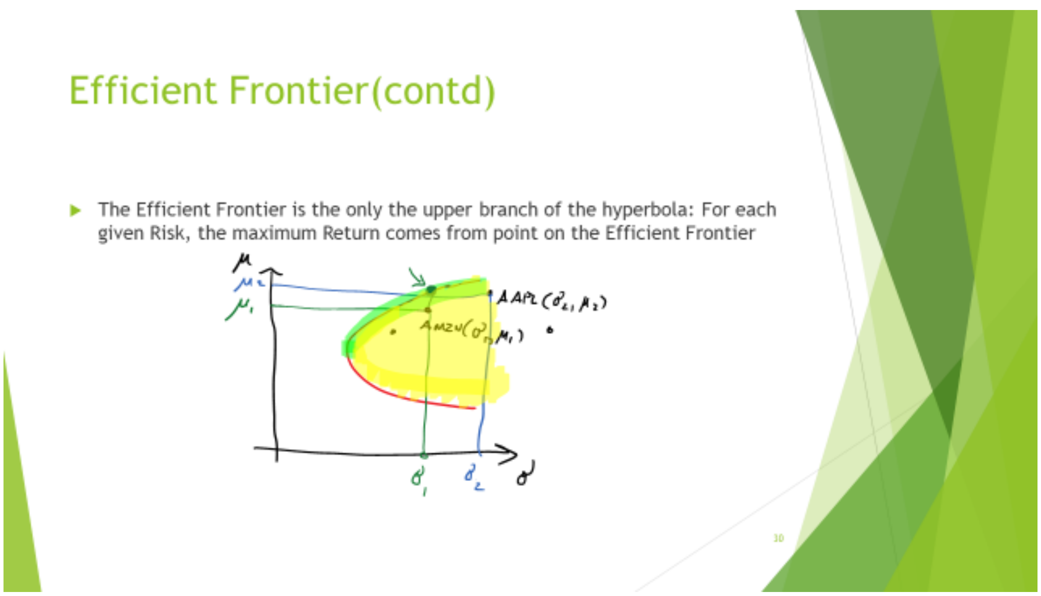
We can,

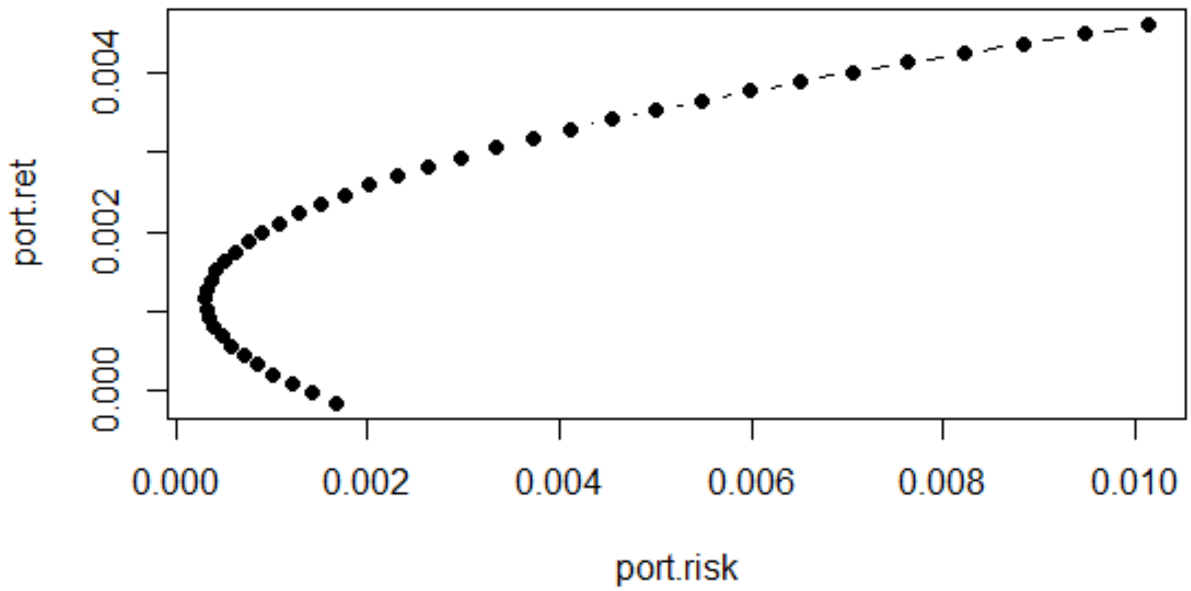
(1) Maximize Expected return, given a known Risk, OR

(2) Minimize Risk, given a known Expected return,

subject to the constraint(*).

$$w_1 \ + w_2 \ + \cdots \ + w_N \ \ = 1 \ (*)$$





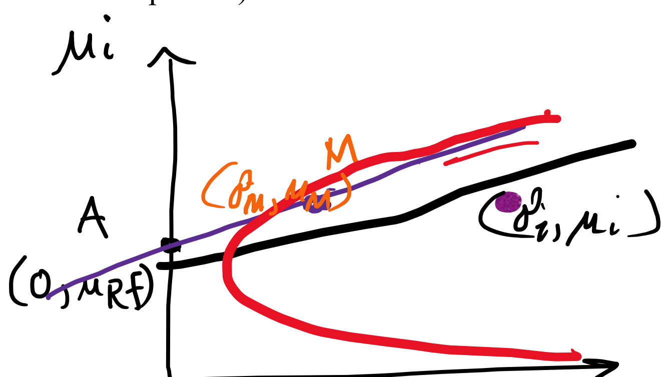
Efficient Frontier of Portfolio with AAPL (Apple) and TSLA (Tesla)

CHAPTER 5: CAPITAL ASSET PRICING MODEL (capm)- MARKET PORTFOLIO

5.1 Capital Market Line (CML), Capital Allocation Line (CAL); Sharpe ratio;

Risk-free asset (Bond) A: is always included in the portfolio, displayed by $A(0, \mu_{Rf})$

- **Market Portfolio M:** (σ_M, μ_M) On the $\sigma - \mu\ plane,$ Market portfolio M will be displayed by . M , $(\sigma_M\ , \mu_M)$, the point of tangency of AM to the Efficient Frontier(Will not be proved)



- The line passing through A, M is called **Capital Market Line (CML)**
- The line passing through A, P is called a Capital Allocation Line (CAL)

σi

(use simple geometry) Slope $=\frac{\mu_i\ -\ \mu_{Rf}}{\sigma_i}$

Relation between $\mu_i,\ \sigma_i$

Equation: $\mu_i\ \ - \mu_{Rf}\ = \left(\frac{\mu_i\ -\ \mu_{Rf}}{\sigma_i}\right)\sigma_i$

- **Sharpe Ratio of an asset : for any asset inside the Feasible region**

Sharpe ratio: Sharpe of the asset $= \frac{\mu_i - \mu_{Rf}}{\sigma_i}$

(1) Optimal asset selection: assets are added to the portfolio using their beta.

(2) The optimal weights: maximize the Sharpe ratio of portfolio to find the optimal weights of assets.

5.2 Beta of an asset

(1) Beta is used to compare different stocks performance;

For any stock i:

$$\beta_i = \frac{Cov(R_i, R_M)}{Var(R_M)} \quad ; \ where\ R_M: \ Market\ portfolio$$

$$\mu_i = \mu_{rf} + \beta_i(\mu_M - \mu_{rf}) \ ;$$
$$0r \quad \mu_i - \mu_{rf} = \beta_i(\mu_M - \mu_{rf})$$

Excess Return of Stock = Beta * Market Excess Return

μ_M : $Market\ average$; μ_{rf}: $riskfree$

(2) Beta can be estimated as coefficient from a **regression model**, using

$R_i - R_{rf}$ $as\ reponse.$

$(R_M - R_{rf})$ $as\ predictor,$

(3) $\mu_i - \mu_{rf} = \beta_i(\mu_M - \mu_{rf})$; $or\ E(R_i) = \mu_{rf} + \beta_i(\mu_M - \mu_{rf})$;

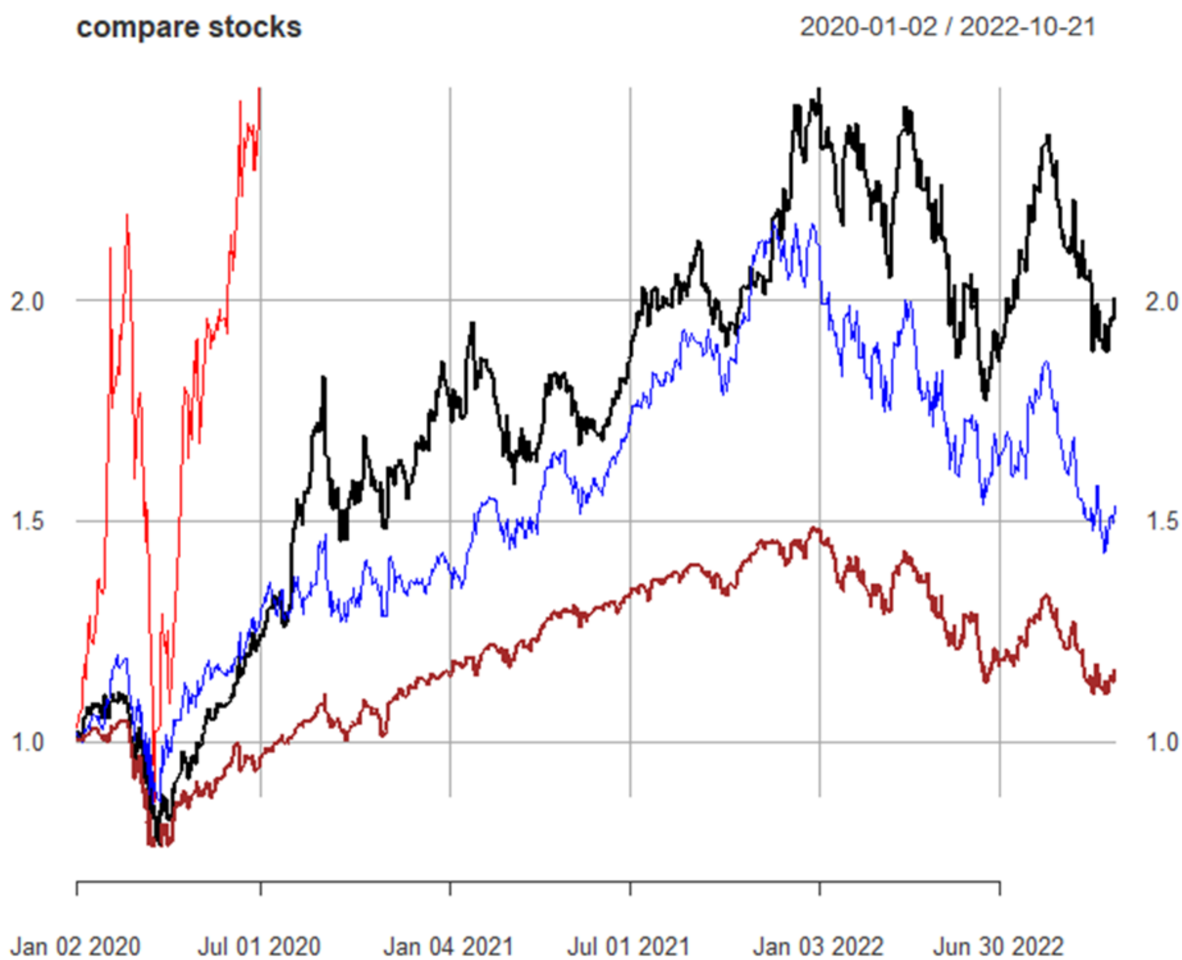
5.3 R Implementation

```
##################################
#### (1) Import stock prices- returns.
#############################################


start<- as.Date("2019-12-31")
end<- as.Date("2022-10-22")
AAPL <-getSymbols("AAPL",from=start, to=end, auto.assign=F)
TSLA <-getSymbols("TSLA",from=start, to=end, auto.assign=F)
MSFT <-getSymbols("MSFT",from=start, to=end, auto.assign=F)
AAPL.ret <- dailyReturn(AAPL$AAPL.Close)
TSLA.ret <- dailyReturn(TSLA$TSLA.Close)
MSFT.ret <- dailyReturn(MSFT$MSFT.Close)



#### combine into an xts-object of all returns
> returns <-cbind(AAPL.ret, TSLA.ret,MSFT.ret)
> colnames(returns)<- c("AAPL.ret", "TSLA.ret", "MSFT.ret")
> head(returns)
           AAPL.ret   TSLA.ret     MSFT.ret
2019-12-31  0.000000000 0.00000000  0.000000000
2020-01-02  0.022816333 0.02851818  0.018516158
2020-01-03 -0.009722044 0.02963325 -0.012451750
2020-01-06  0.007968248 0.01925466  0.002584819
2020-01-07 -0.004703042 0.03880052 -0.009117758
2020-01-08  0.016086289 0.04920484  0.015928379

#######################################
###### Risk-free rate  and Market Portfolio return
#####################################################

####Risk-free rate

 rf<-getSymbols("DGS1",src="FRED" ,auto.assign = F)
dim(rf)
```

```
[1] 15866     1

rf1 <- rf[ date >= "2019-12-31"  & date<="2022-10-25",  ]


##suppose risk-free rate as the average for the period
rf1 <- na.omit(rf1)
 riskfree <-mean(rf1)/100
riskfree
[1] 0.008568458


## Market portfolio
benchmark<-getSymbols.yahoo("^GSPC", from=start, to=end, periodicity =
"daily",auto.assign=F)[  ,4]

bench.return <- dailyReturn(benchmark)




################################
## beta's
################################
beta.AAPL
          daily.returns
daily.returns     1.171948


 ## beta for TSLA

 ## beta for TSLA
beta.TSLA<- cov(bench.returns,  returns.TSLA)/var(bench.returns)
 beta.TSLA
          daily.returns
daily.returns     1.457297


## beta for MSFT
 beta.MSFT<- cov(bench.returns,  returns.MSFT)/var(bench.returns)
beta.MSFT
          daily.returns
daily.returns     1.153356
```

REMARK: TSLA has largest Beta. Beta>1.0 shows the stock perform better than the Market, at least 1.5 the Market.

```
#################################################
##Plot the Cumulative Product of (1+ R(t))
##################################################
cumprod<- cumprod(1+ returns[-1, ])
lines(cumprod$TSLA.ret, type="l", col="red")
lines(cumprod$MSFT.ret, type="l", col="blue")
legend("bottom.right", c("TSLA: red- AAPL: black - MSFT: blue"))
```



References

1. Ashar, Jay. "What are the different financial models?". *Financial Director*, What are the different financial models? - Financial Director. Accessed 01 March 2022.
2. Capinski, Marek and Zastawniak, Tomasz. *Mathematics For Finance: An Introduction to Mathematical Engineering*. Springer, 2003.
3. Copeland, Weston, Shastri. *Financial Theory and Corporate Policy*. 2005.
4. Jack Clark Francis, Dongcheon Kim. *Modern Portfolio Theory*.
5. *Investopedia*. IAC & Dotdash Meredith, 1999, June 6.

   https://www.investopedia.com/. Accessed 20 February 2022.

6. Little, Ken. "USE ALL FOUR ASSET CLASSES to BUILD YOUR PORTFOLIO".

   the balance, Four Asset Classes to Build Your Portfolio (thebalance.com). Accessed

   16 February 2022.

7. Thakar, Chainika. "Introduction to Quantitative Finance." Introduction to

   Quantitative Finance (quantinsti.com). Accessed 20 February 2022.

Acknowledgement

Name of Advisor: Ha Nguyen PhD

Thanks to
Committe member: Dr. Young-Sha Chan
Course coordinator: Dr. John King