Question 1

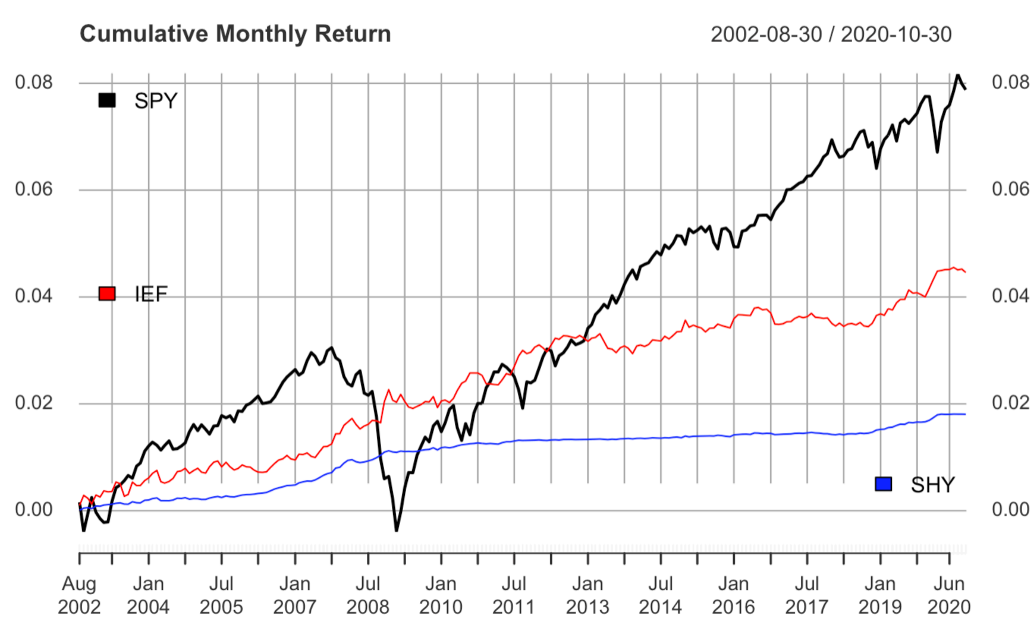
1.

```
[1] "mean return in annual basis"
       SPY        IEF        SHY
0.09114246 0.05070245 0.02069700
[1] "standard deviation in annual basis"
       SPY        IEF        SHY
0.19430583 0.06590418 0.01390208
```

Compared each annually mean returns with bond ETFs, it is easy to find that SPY has higher return than other two bonds ETFs. It is because SPY has higher votility than bonds ETFs, that is, the higher sigma. In terms of risk-return reward trade-off, if a person is a risk taker, he can choose SPY. However, if he is a risk aversion, he might be more suitable to bonds ETFs.

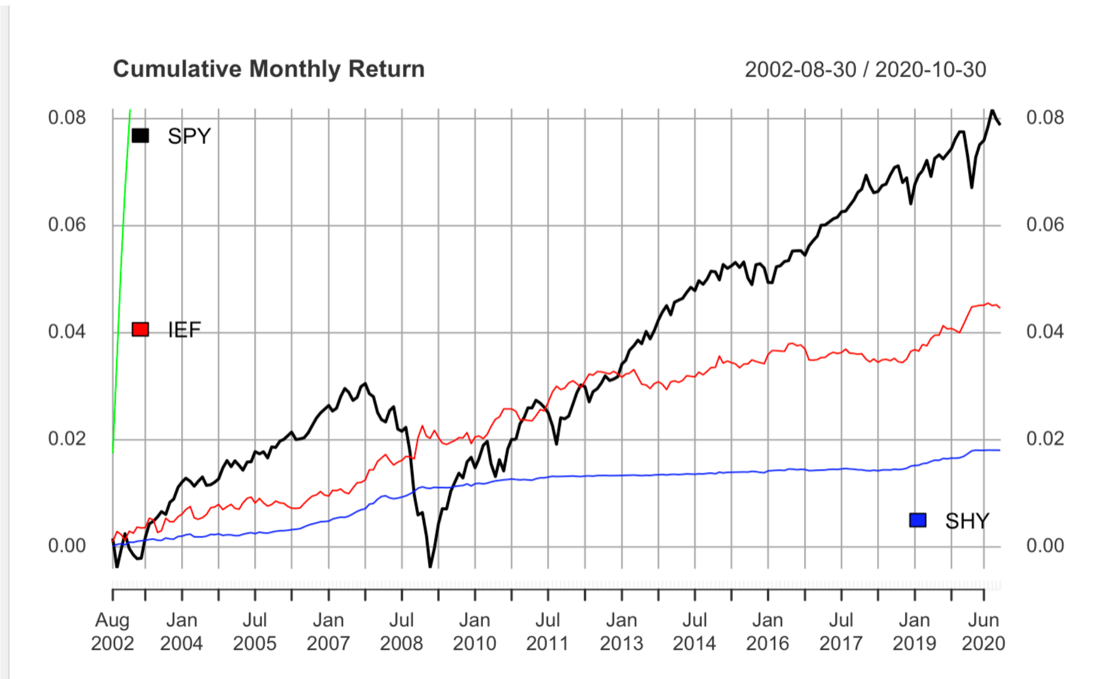
2.

	SPY	IEF	SHY
SPY	1.0000000	-0.3248029	-0.3518223
IEF	-0.3248029	1.0000000	0.7704028
SHY	-0.3518223	0.7704028	1.0000000

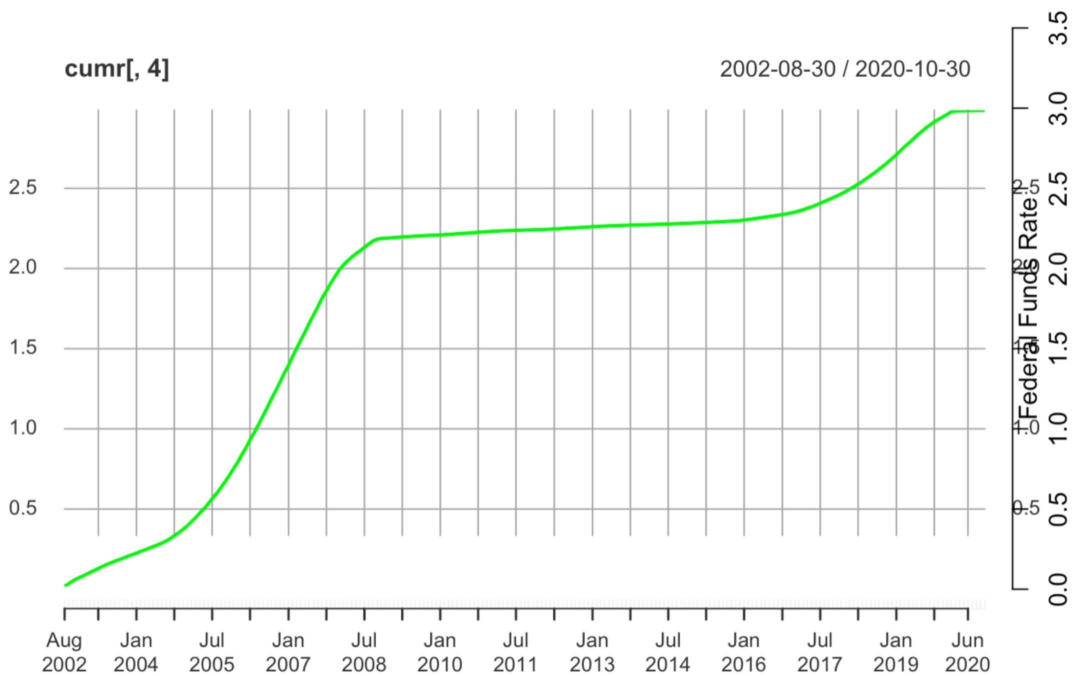


The plot shows that SPY (the black line) has the opposite monthly return with IEF and SPY (the red and blue line). That is, when the return of SPY has positive return IEF and SPY will have negative return.

3(a).



Cumulative Monthly Return
2002-08-30 / 2020-10-30
SPY
IEF
SHY
0.08
0.06
0.04
0.02
0.00
Aug 2002
Jan 2004
Jul 2005
Jan 2007
Jul 2008
Jan 2010
Jul 2011
Jan 2013
Jul 2014
Jan 2016
Jul 2017
Jan 2019
Jun 2020



cumr[, 4]
2002-08-30 / 2020-10-30
2.5
2.0
1.5
1.0
0.5
Federal Funds Rate
0.0
3.0
3.5
Aug 2002
Jan 2004
Jul 2005
Jan 2007
Jul 2008
Jan 2010
Jul 2011
Jan 2013
Jul 2014
Jan 2016
Jul 2017
Jan 2019
Jun 2020

(b)

Call:

lm(formula = dt ~ MuR[, 1])

Coefficients:

(Intercept)   MuR[, 1]

 -0.0001327   0.1594927

Call:

lm(formula = dt ~ MuR[, 2])

Coefficients:

(Intercept)   MuR[, 2]

 -1.978e-05   -2.821e-01

Call:

lm(formula = dt ~ MuR[, 3])

Coefficients:

(Intercept)   MuR[, 3]

  0.0001402   -2.6297579

Beta of each regression: 0.1594927, -2.821e-01 , -2.6297579

Question 2

1.We have the valuation function:

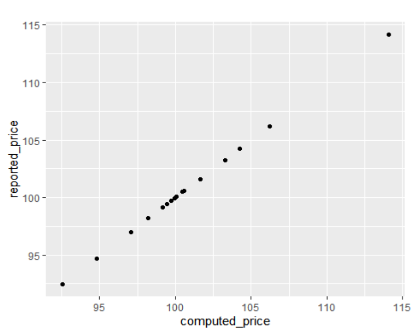
```
bond <- function(x,output){
  cf <- c(rep(x[1] * x[2], x[3] - 1), x[1] * (1 + x[2]))
  cf <- data.frame(cf)
  cf$t <- as.numeric(rownames(cf))
  cf$pv_factor <- 1 / (1 + x[4])^cf$t
  cf$pv <- cf$cf * cf$pv_factor
  sum(cf$pv)
}
```

We can get the output of computed value

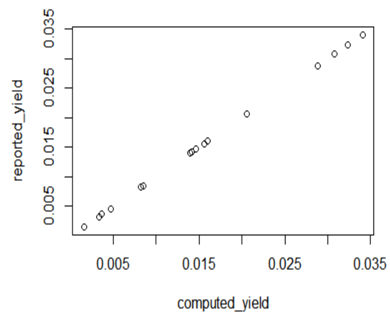
```
computed_price <- apply(x,1,bond)
```

With the reported price， we can plot the relationship

```
library(ggplot2)
ggplot(data = NULL, aes(x = computed_price, y = reported_price)) +
  geom_point()
```



Q1



Q2

2.

Input the new data. Use a while loop and a Uniroot function to get the result, which is saved in computed_yield

```
token = 1
computed_yield <- list()
while(token < 17){
# Create cash flow vector
cf <- c(-x[token,1],rep(x[token,2] * x[token,3], x[token,4]-1), x[token,2] * (1 + x[token,3]))
# Create bond valuation function
bval <- function(i, cf, t=seq(along = cf))sum(cf / (1 + i)^t)
# Create ytm() function using uniroot
ytm <- function(cf) {uniroot(bval, c(0, 1), cf = cf)$root}
# Use ytm() function to find yield
computed_yield[[token]] <- ytm(cf)
# New loop
token <- token + 1
}
```

With the reported yield:

```
reported_yield = c(0.0288,0.0307,0.0323,0.0340,0.0147,0.0140,0.0156,0.0206,0.0031,0.0046,0.0085,0.0142,0.0015,0.0036,0.0082,0.0160)
```

We can plot the relationship

```
plot(x = computed_yield, y = reported_yield)
```

```
3.
token = 1
MD = matrix(nrow = 4,ncol = 4)
while(token < 17){
 cf <- c(rep(x[token,1] * x[token,2], x[token,3]-1), x[token,1] *(1 + x[token,2]))
 cf <- data.frame(cf)
 cf$t <- as.numeric(rownames(cf))
 cf$pv_factor <- 1 / (1 + x[token,4])^cf$t
 cf$pvt <- cf$cf * cf$pv_factor * cf$t
 MD[[token]] <- sum(cf$pvt/computed_price[[token]])
 token <- token + 1
}
print(MD)
```

```
##          2018      2019 March 2020  Nov 2020
## [1,]  1.973203  1.985226   1.988916  1.998701
## [2,]  4.726402  4.854796   4.891613  4.975043
## [3,]  8.808622  9.314604   9.383758  9.719726
## [4,] 19.758713 22.318520  23.377183 24.593082
```

For a clearer view, I decided to combine the 4 tables together.

We can see that the bond issued in 2018 has the shortest Macaulay duration and Nov 2020 has the longest Macaulay duration. It is because the yield is far less than the coupon in bond March 2020. Macaulay duration reflects the estimated years that the holder can get back his money. If it is shorter than the maturity, it means the cash flow worth more.

4.We compute the modified duration first

```r
print(ModD)
```

```
##          2018      2019 March 2020  Nov 2020
## [1,]  1.917966  1.956466   1.982769  1.995708
## [2,]  4.585623  4.787767   4.869214  4.957197
## [3,]  8.533006  9.171528   9.304668  9.640672
## [4,] 19.109006 21.868038  23.049874 24.205789
```

Then we compute the convexity

```r
print(Convexity)
```

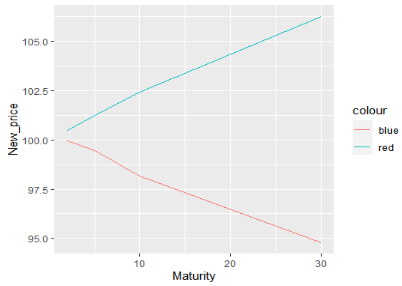
```
##           2018       2019 March 2020   Nov 2020
## [1,]   5.567506   5.770019   5.918909   5.976862
## [2,]  26.186858  28.049494  28.867209  29.587024
## [3,]  86.997023  96.950435  99.292005 104.184809
## [4,] 488.282953 594.970650 642.261469 688.316705
```

Set the yield rate of panel(d) down by 25bp, we have

```r
token = 1
computed_price25 = matrix()
while(token < 5){
  computed_price25[[token]] <- 100 * (1 + ModD[token,4]*0.0025 + 0.5 * Convexity[token,4]*(0.0025^
2))
  token <- token + 1
}
print(computed_price25)
```

```
## [1] 100.5008 101.2485 102.4427 106.2665
```

Plot together



The original price goes down with the increase of maturity because the yield exceeds the coupon rate so that the cash flow of each term is below the expectation of investor. The longer the maturity is , the lower the price of bond should be

But once we have the yield come down 25bp, things reverse, the coupon rate exceeds the yield rate so the investor get coupon more than they expected every term, so the bond worth more.

5.

```r
#Allocate x to 13, (100000-99.95*x)/99.45 to 14
#Combine the cash flow
x <- data.frame(p = c(100,100),r = c(0.0013,0.0025),ttm = c(2,5),y = c(0.0015,0.0036))

 cf_13 <- c(rep(x[1,1] * x[1,2], x[1,3]-1), x[1,1] *(1 + x[1,2]),0,0,0)
 cf_13 <- data.frame(cf_13)
 cf_13$t <- as.numeric(rownames(cf_13))
 cf_13$pv_factor <- 1 / (1 + x[1,4])^cf_13$t
 cf_13$pvt <- cf_13$cf_13 * cf_13$pv_factor * cf_13$t

 cf_14 <- c(rep(x[2,1] * x[2,2], x[2,3]-1), x[2,1] *(1 + x[2,2]))
 cf_14 <- data.frame(cf_14)
 cf_14$t <- as.numeric(rownames(cf_14))
 cf_14$pv_factor <- 1 / (1 + x[2,4])^cf_14$t
 cf_14$pvt <- cf_14$cf_14 * cf_14$pv_factor * cf_14$t

 fun <- function(x){sum(cf_13$pvt * x + cf_14$pvt * (100000-99.95*x)/99.45)/100000 -3 }
 uniroot(fun,lower = 0,upper = 1000)
```

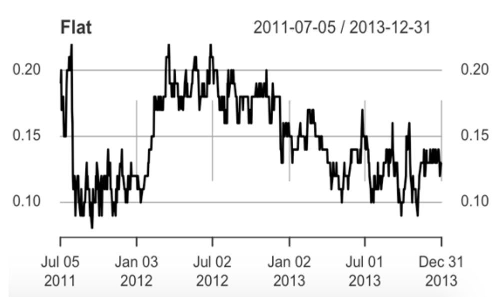
```
## $root
## [1] 663.9911
```

I would have at least 664 units of bond 13 to maintain the portfolio duration to below 3 years.

For targeting a duration of 6, I think its impossible because bond 13 and 14 have duration of 2 and 5, so that the portfolio duration must be within this range.

Question 3

1.



2.

	DGS1MO	DGS3MO	DGS1	DGS2	DGS5	DGS7	DGS10
Mean	1.253231	1.307046	1.520943	1.755719	2.439388	2.798940	3.123090
Sd	1.468261	1.490872	1.488377	1.415346	1.271423	1.207685	1.166134
Skewness	1.242960	1.209582	1.044050	0.845418	0.382167	0.179529	0.010124
Kurtosis	0.606514	0.489072	0.076835	-0.370926	-0.876428	-0.952075	-0.979869

```
> basicStats(yield$DGS1MO)
                 DGS1MO
nobs        4838.000000
NAs            0.000000
Minimum        0.000000
Maximum        5.270000
1. Quartile    0.070000
3. Quartile    1.900000
Mean           1.253231
Median         0.860000
Sum         6063.130000
SE Mean        0.021109
LCL Mean       1.211847
UCL Mean       1.294614
Variance       2.155791
Stdev          1.468261
Skewness       1.242960
Kurtosis       0.606514
```

```
> basicStats(yield$DGS3MO)
                 DGS3MO
nobs        4838.000000
NAs            0.000000
Minimum        0.000000
Maximum        5.190000
1. Quartile    0.100000
3. Quartile    1.947500
Mean           1.307046
Median         0.930000
Sum         6323.490000
SE Mean        0.021434
LCL Mean       1.265026
UCL Mean       1.349067
Variance       2.222700
Stdev          1.490872
Skewness       1.209582
Kurtosis       0.489072
```

```
> basicStats(yield$DGS1)
                   DGS1
nobs        4838.000000
NAs            0.000000
Minimum        0.080000
Maximum        5.300000
1. Quartile    0.240000
3. Quartile    2.310000
Mean           1.520943
Median         1.180000
Sum         7358.320000
SE Mean        0.021398
LCL Mean       1.478992
UCL Mean       1.562893
Variance       2.215265
Stdev          1.488377
Skewness       1.044050
Kurtosis       0.076835
```

```
> basicStats(yield$DGS2)
                   DGS2
nobs        4838.000000
NAs            0.000000
Minimum        0.110000
Maximum        5.290000
1. Quartile    0.580000
3. Quartile    2.610000
Mean           1.755719
Median         1.360000
Sum         8494.170000
SE Mean        0.020348
LCL Mean       1.715827
UCL Mean       1.795611
Variance       2.003203
Stdev          1.415346
Skewness       0.845418
Kurtosis      -0.370926
```

```
> basicStats(yield$DGS5)
                   DGS5
nobs        4838.000000
NAs            0.000000
Minimum        0.190000
Maximum        5.230000
1. Quartile    1.500000
3. Quartile    3.357500
Mean           2.439388
Median         2.200000
Sum        11801.760000
SE Mean        0.018279
LCL Mean       2.403553
UCL Mean       2.475224
Variance       1.616517
Stdev          1.271423
Skewness       0.382167
Kurtosis      -0.876428
```

```
> basicStats(yield$DGS7)
                   DGS7
nobs        4838.000000
NAs            0.000000
Minimum        0.360000
Maximum        5.290000
1. Quartile    1.900000
3. Quartile    3.800000
Mean           2.798940
Median         2.670000
Sum        13541.270000
SE Mean        0.017363
LCL Mean       2.764901
UCL Mean       2.832979
Variance       1.458504
Stdev          1.207685
Skewness       0.179529
Kurtosis      -0.952075
```

```
> basicStats(yield$DGS10)
                  DGS10
nobs        4838.000000
NAs            0.000000
Minimum        0.520000
Maximum        5.440000
1. Quartile    2.190000
3. Quartile    4.160000
Mean           3.123090
Median         2.955000
Sum        15109.510000
SE Mean        0.016765
LCL Mean       3.090222
UCL Mean       3.155958
Variance       1.359869
Stdev          1.166134
Skewness      -0.010124
Kurtosis      -0.979869
```

3.

According to the table, it shows that when the Sd becomes higher, the Kurtosis also increases. So Sd is in direct ratio to Kurtosis.

4.

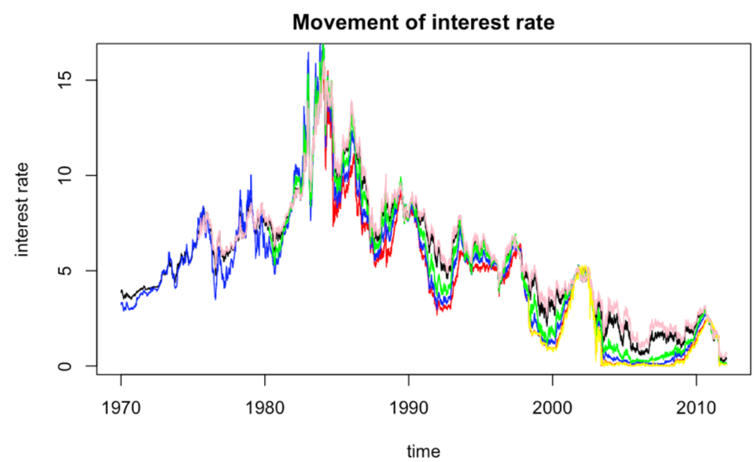
If we control for interest rate regimes, where you round the 1-month yields to the closest 0.25unit. The volatility of the yield will increase. As well as convexity which is a measure of the degree of curvature of the bond price curve varies with the slope of the bond price-yield curve and the yield. So the convexity will also increase with the volatility of the yield. The greater the convexity, the greater the degree of curvature of the bond price curve, and the greater the error produced by the interest rate risk measured by the modified duration. The greater the convexity of the yield, the greater the interest rate risk of the bond.

Part 2

1) From the following chat, we can see that Interest rates on bonds of different maturities move together over time.

```{r}
bonds = data.frame(na.approx(cbind(P,P1,P2,P3,P4,P5,P6,P7)))

x <- as.Date(index(bonds))
plot(x,bonds[,6],type="l",xlab="time",ylab="interest rate",col="black",main="Movement of interest rate")
lines(bonds[,3],col="red")
lines(bonds[,4],col = 'blue')
lines(bonds[,5],col = 'green')
lines(bonds[,2],col = 'yellow')
lines(bonds[,7],col = 'pink')
```

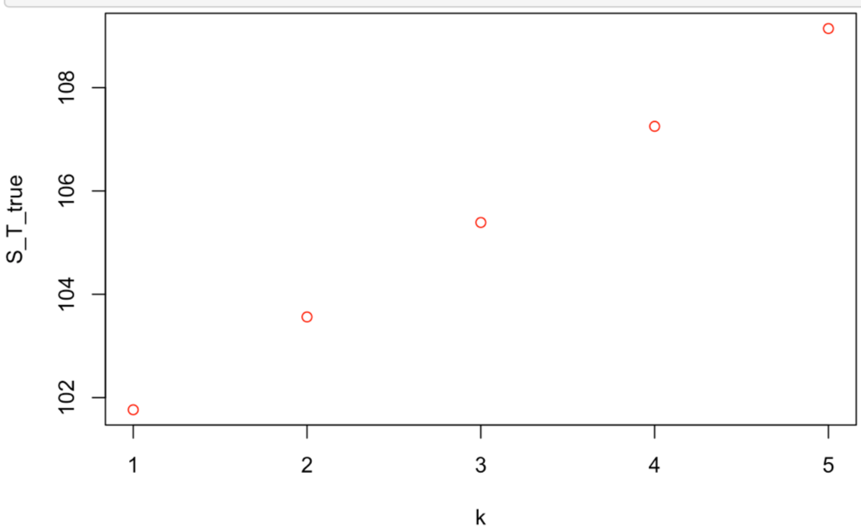


Question 4

1.

```
S = 100
r = 0.0175
sig = 0.2
tau = 1
N = 10 ^ 3
k = seq(1, 5, by = 1)

S_T_true = S * exp(k * r)
plot(S_T_true~k,col="red")
```



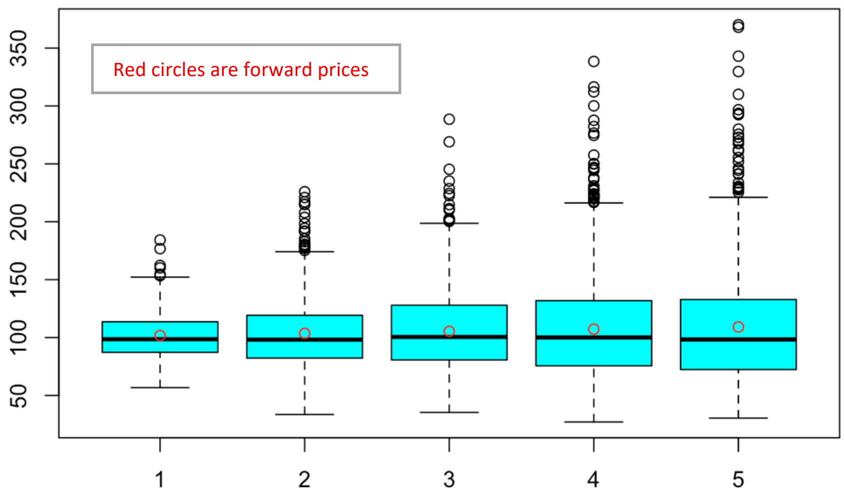
The graph shows that the difference of the log returns should be proportional with the time to maturity.

2.

```
sim_f = function(tau) {
  R_tau = rnorm(N, (r - (sig ^ 2) / 2) * tau, sqrt(tau) * sig)
  S_T = S * exp(R_tau)
  return(S_T)
}

S_T_sim = sapply(k, sim_f)
S_T_seq=apply(S_T_sim,2,mean)

boxplot(S_T_sim, col = "5")
points(S_T_true ~ k, col = "red")
```



The graph shows that the expectation of the simulation almost overlaps with the forward contract price which implies that under no-arbitrage, there should be no difference between buying the asset today and investing in risk free rate market and buying the forward contract and taking the delivery.

3.

Since I currently have a long position in the stock index, I would short the 1-year forward contract to hedge which is same as invest at risk free rate.

```
temp = S_T_sim - S_T_true
VaR = mean(temp) - quantile(temp,0.05)
```

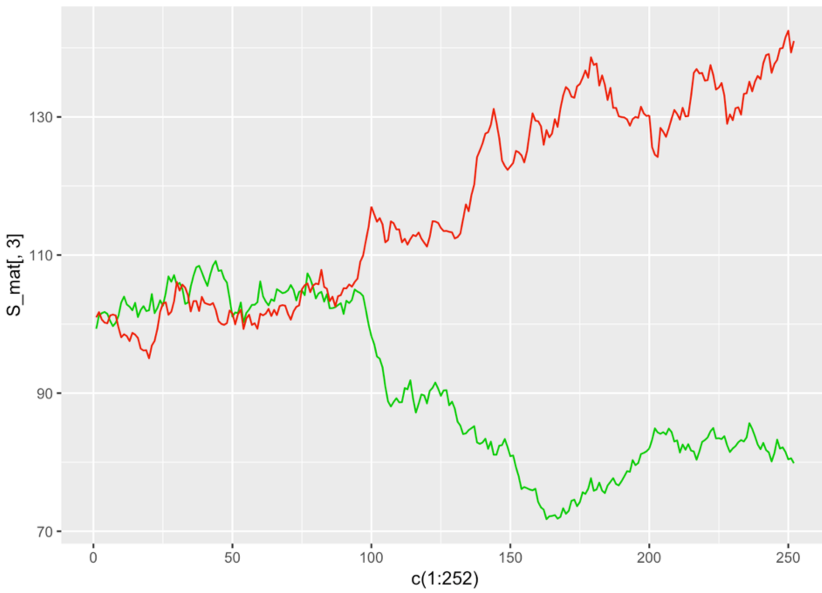
$PL = S_t - S_0 e^{r\tau}$, 95% VaR is 50.57.

4.

```
dt = 1 / 252
day = tau / dt
sim_path = function(x) {
  R_tau = rnorm(day, (r - (sig ^ 2) / 2) * dt, sqrt(dt) * sig)
  S_T = S * exp(cumsum(R_tau))
  return(S_T)
}

S_mat=sapply(1:10, sim_path)

ggplot(as.data.frame(S_mat[,c(3,5)]),aes(x=c(1:252))) +
  geom_line(aes(y=S_mat[, 3]),col="green3") +
  geom_line(aes(y=S_mat[, 5]),col="red2")
```



Long stock price case,

- Short sell the asset today and invest in stock at $100.
- At maturity, receive $141.00 and purchase the asset at $S_0 e^{r\tau}$.
- We get 39.48.

Short stock price case,

- Borrow S_0 value of stock and sell to get money, then buy asset at spot price to enter forward contract.
- At maturity, purchase the borrowed stock; deliver the asset and receive $S_T - 0.25$.
- We get $21.66.

5.

Since after 6 months, the rate changes to 0.0125, the value of the forward contract changes to $S_0 e^{0.175\times0.5} e^{0.125\times0.5} = 0.254$. As a result, my arbitrage strategy will loss $-(0.25 - 0.254) = 0.004$.

Question 5

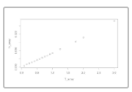
1(a): We use the mid_price of quote bid and ask as the price of the forward price, and based on the interest rate parity, get the θ equals to 0.9007% = 90.07bp

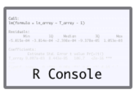
```
According to interest rate parity:
$$
F_0(\tau) = S_0 e^{\theta \tau}
$$
```

$$F_0(\tau) = S_0 e^{\theta \tau}$$

```
We use the mid_price of quote bid and ask as the price of the foward price.
```

~~~~
```{r}
S0 = 1.1836
F_t = S0 + 0.5*(quote_ask_array + quote_bid_array)/10000
ln_array = log(F_t/S0)
plot(T_array, ln_array)
linear_model = lm(ln_array~T_array - 1)
summary(linear_model)
```
~~~~






```
Call:
lm(formula = ln_array ~ T_array - 1)

Residuals:
       Min         1Q     Median         3Q        Max 
-5.015e-04 -3.814e-04 -2.398e-04 -9.378e-05  1.053e-03 

Coefficients:
         Estimate Std. Error t value Pr(>|t|)    
T_array 9.007e-03  8.443e-05   106.7   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

1(b): Annualized sigma is 0.07255

~~~~
```{r}
history_rate = getSymbols("EURUSD=X", from ='2016-01-01', to='2020-04-03', auto.assign = FALSE)[,6]
log_r = na.omit(log(history_rate/lag(history_rate)))
sigma = sd(log_r)*sqrt(252)
sigma
```
~~~~

```
EURUSD=X contains missing values. Some functions will not work if objects contain missing values in the middle of the series.
Consider using na.omit(), na.approx(), na.fill(), etc to remove or replace them.[1] 0.07255397
```

```
The annulized sigma is 0.07255.
```

2. The 99% VaR of the exporter's P&L is 220706.2.

```
$$
S_\tau = S_0 exp((\theta-\frac{\sigma^2}{2})\tau + \sigma B_\tau)
$$
```

$$
S_\tau = S_0 exp((\theta - \frac{\sigma^2}{2})\tau + \sigma B_\tau)
$$

```{r}
MC_simulation <- function(S0,theta,sigma,t,simu_times){
  B_t = rnorm(simu_times, mean=0,sd=t^0.5)
  S_t = S0*exp((theta - 0.5*sigma^2)*t+sigma*B_t)
  V_t = S_t - S0
  return(V_t)
}


VaR <- function(S0,theta,sigma,t,simu_times, alpha){
  V_t_array = MC_simulation(S0,theta,sigma,t,simu_times)
  EV_t = mean(V_t_array)
  Q = quantile(V_t_array, c(alpha), TRUE)
  return(EV_t - Q)
}

VaR(1.1836, 90.07/10000, sigma, 10.5/12, 10000000, 0.01)*1250000
```

```
      1%
220706.2
```

3.

(a) The 99% VaR of the P&L with unitary hedging is 471.5155.

```{r}
MC_simulation2 <- function(S0,theta,sigma,t,td,simu_times){
  B_t = rnorm(simu_times, mean=0,sd=t^0.5)
  S_t = S0*exp((theta - 0.5*sigma^2)*t+sigma*B_t)
  F_0 = S0*exp(theta*(td-0))
  F_t = S_t*exp(theta*(td-t))
  V_t = S_t - S0 + F_0 - F_t
  return(V_t)
}


VaR2 <- function(S0,theta,sigma,t,td, simu_times, alpha){
  V_t_array = MC_simulation2(S0,theta,sigma,t,td,simu_times)
  EV_t = mean(V_t_array)
  Q = quantile(V_t_array, c(alpha), TRUE)
  return(EV_t - Q)
}
```

```{r}
VaR2(1.1836, 90.07/10000, sigma, 10.5/12,13/12,100000000, 0.01)*1250000
```

```
      1%
471.5155
```

(b) The new 99% VaR of the P&L now is 51027.9, which is much higher than (a).

Basis risk remains because of differing maturities. The new futures contract expiring in September 2020, which is before the delivery. The basis risk is the exchange rate change after the expire day of futures (sep-mid 2021)

```{r}
MC_simulation3 <- function(S0,theta,sigma,t,td,simu_times){
  B_t1 = rnorm(simu_times, mean=0,sd=td^0.5)
  B_t2 = B_t1 + rnorm(simu_times, mean=0,sd=(t-td)^0.5)
  S_t1 = S0*exp((theta - 0.5*sigma^2)*td+sigma*B_t1)
  S_t2 = S0*exp((theta - 0.5*sigma^2)*t+sigma*B_t2)
  F_0 = S0*exp(theta*(td-0))
  F_t = S_t1
  V_t = S_t2 - S0 + F_0 - F_t
  return(V_t)
}


VaR3 <- function(S0,theta,sigma,t,td, simu_times, alpha){
  V_t_array = MC_simulation3(S0,theta,sigma,t,td,simu_times)
  EV_t = mean(V_t_array)
  Q = quantile(V_t_array, c(alpha), TRUE)
  return(EV_t - Q)
}
```

```{r}
VaR3(1.1836, 90.07/10000, sigma, 10.5/12,10/12,100000000, 0.01)*1250000
```

```
      1%
 51027.9
```

4.

1) EUFX (ProShares Short Euro ETF)

It provides inverse exposure to the European currency and is designed to provide 100% of the inverse, or opposite, return of the U.S. dollar price of the euro, on a daily basis. It could be used to hedge against the risk that price of European currency will fall.

2) EUO (ProShares UltraShort Euro ETF)

Similar to EUFX, EUO provides inverse exposure to the European currency. It provides 200% of the inverse return of the U.S. dollar price of the euro on a daily basis

3) DRR (Market Vectors Double Short Euro ETN)

Similar to EUO, DRR tracks the Double Short Euro Index, which also provides a negative 200% exposure to the euro.

4) ULE (ProShares Ultra Euro)

Different from EUFX, EUO, and DRR, ULE long European currency, and provides 200% exposure to the euro. It could be used to hedge against the risk that price of European currency will increase.

5) FXE (Invesco Currency Shares Euro Currency Trust)

This ETF offers exposure to the euro, the official currency of the eurozone, relative to the U.S. dollar, increasing in value when the euro strengthens and declining when the dollar appreciates. This fund could be appropriate for investors seeking to hedge exchange rate exposure or bet against the greenback. For investors seeking exposure to the EUR/USD exchange rate, FXE is the only real ETF option available.