STATS 509 HOMEWORK 1 YUAN YIN

Problem 1

(a) $h(X) = X^3 + 2$ is a strictly increasing function, thus $Y_q = h(X_q) = 47.2361$

(b) $h(X) = \exp(-X)$ is a strictly decreasing function, thus $Y_q = h(X_{1\text{-}q}) = 4.773611$

Problem 2

As we know, net return: $R_t = X_t / X_{t-1} - 1$, and log-return: $r_t = \log(X_t / X_{t-1})$. Thus, we can conclude that $R_t = \exp(r_t) - 1$.

It's an increasing function of r_t. Thus we have $R_q = \exp(r_q) - 1$

(a) $\widetilde{\text{VaR}} = 0.06638818$, VaR $= 6638818$

(b) We can compute λ for each case according to the equation $SD(X) = \lambda\sqrt{\upsilon/(\upsilon-2)}$

When $\upsilon = 12$, $\lambda = 0.02282177$, $\widetilde{\text{VaR}} = 0.07317694$, VaR $= 7317694$;

When $\upsilon = 6$, $\lambda = 0.02041241$, $\widetilde{\text{VaR}} = 0.08125373$, VaR $= 8125373$;

When $\upsilon = 3$, $\lambda = 0.01443376$, $\widetilde{\text{VaR}} = 0.09602660$, VaR $= 9602660$

Problem 3

(a) First we plot the adjusted closing price as a function of time as follows:

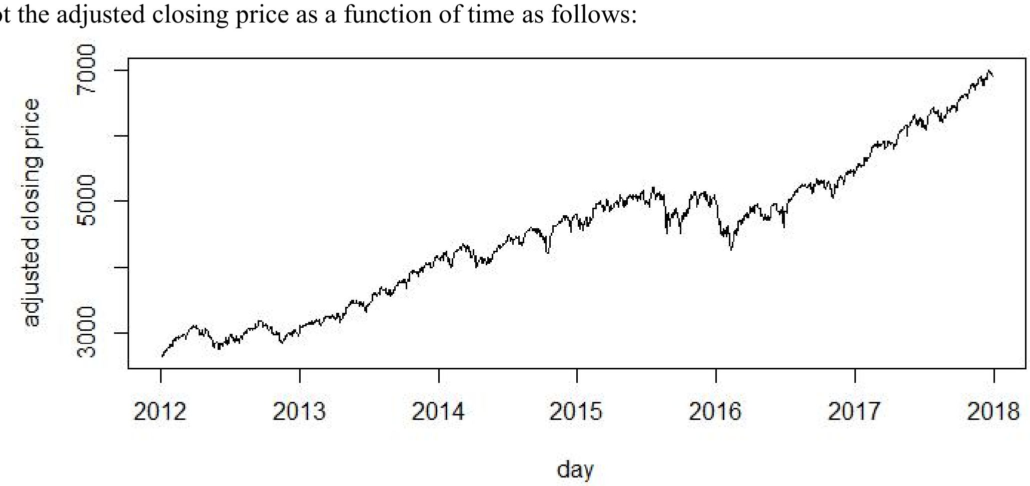

Fig 1 adjusted closing price vs time

Then let's plot the log returns based on adjusted closing price as a function of time:

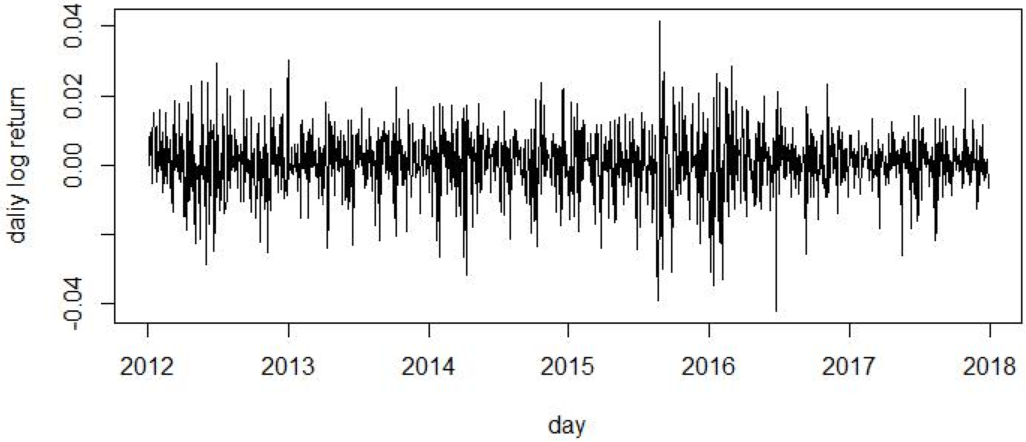

Fig 2 log returns vs time

As we can see above, generally, the adjusted closing price is increasing steadily as time goes by, but there are two different time points which is at the end of 2015 and at the beginning of 2016, the adjusted closing price decreased twice obviously.

From figure 2 we can see that the log returns is with mean around 0 and standard deviation no more than 0.04. and the largest variation happens around year 2016, which is corresponding with two decreasing points of figure 1.

(b) For exploring more information about log returns, we first check the summary statistics:

```
> summary(acp_lreturn)
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
-0.0420232 -0.0035846  0.0008784  0.0006352  0.0055978  0.0415203 
```

we can see that the mean of log returns is close to 0, and the data of log returns is close to a symmetry distribution with mean close to zero.

Next let's look at the skewness and kurtosis of log returns:

Skewness = -0.4036703, which means the log returns is a little left skew and we compute kurtosis = 2.002712 > 0, which means the distribution is leptokurtic, it has heavier tails than normal distribution.

Finally let's see the histogram and box-plot of log returns:

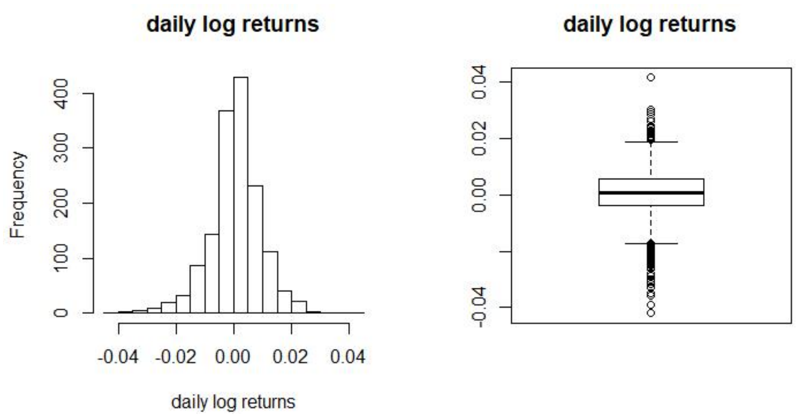

Fig 3 histogram and box-plot of log returns

We can find that from the histogram, the distribution is very like with normal distribution but with higher peak. And from box-plot we can see that the variation of data isn't large, most data focus between [-0.01, 0.01], and there are more data in tails than normal distribution.

(c) As the conclusion we get in problem (b), the kurtosis = 2.002712 > 0, which means the kurtosis of our log returns data is larger than normal distribution, the kurtosis represents the degree of steep of the distribution, thus our data is steeper than normal distribution, which means the tails of log returns is heavier than normal distribution and the peak is higher. On the other hand, the kurtosis of double exponential distribution is 3, which is larger than the kurtosis of our data, this implies that the tails of log returns is lighter than double exponential distribution and the peak is lower than double exponential distribution.

Problem 4

Question 12: the plot is as follows:

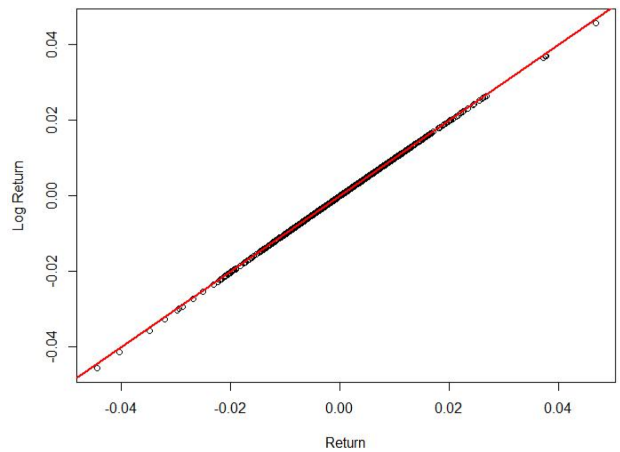

Fig 4 log return vs return

As we can see, the red line is the line “y=x”, and the points of log return are too close to the line to tell them apart, so we can say the two returns are approximately equal.

Question 13: compute the mean and variation of two returns:

Mean of return = 0.0005027479 and mean of log return = 0.0004630553

Standard deviation of return = 0.008900319 and standard deviation of log return = 0.008901467

Comparing them with the first two moments of these two random variable, we can see that they are the same with at least 3 digits. Thus, it’s reasonable to say that they are the same.

Question 16:

To simulate the result with Monte Carlo method, we compute the mean of pay off is 0.0605 > 0, also the 95% confidence interval is [0.05582723, 0.06517277] where zero is not contained in it, which means we are more possible to make profit in this bet. Thus we should make this bet.

Question 17:

From the question we can obviously find that in this new bet, we can only make more profit than the previous bet, we simulate with Monte Carlo method and the mean pay off is 0.243, which is much larger than previous bet, also, the 95% confidence interval is [0.2345938 0.2514062] where 0 is not contained.

Appendix

```
# question 1
q1 = qnorm(0.9, mean = 1, sd = 2)
q1^3 + 2
q2 = qnorm(0.1, mean = 1, sd = 2)
exp(-q2)

#question 2
qa = qnorm(0.003, mean = 0, sd = 0.025)
VaR_tilde = -(exp(qa)-1)
VaR_tilde
VaR = 100 * 10^6 * VaR_tilde
VaR
v = 12
## v = 6 v = 3
lamda = 0.025/sqrt(v/(v-2))
qa = qt(0.003, v) * lamda
VaR_tilde = -(exp(qa)-1)
VaR = 100 * 10^6 * VaR_tilde
#question 3
dat = read.csv("Nasdaq_daily_Jan1_2012_Dec31_2017.csv", header = T)
t = as.Date(dat$Date, format = "%m/%d/%Y")
idx1 = which(is.na(dat$Adj.Close) == FALSE)
times = t[idx1]
acp = dat$Adj.Close[idx1]
acp_lreturn = diff(log(acp))
plot(times, acp, xlab = 'day', ylab = 'adjusted closing price', type = 'l')
plot(acp_lreturn, xlab = 'day', ylab = 'daliy log return', type = 'l')
```

```
summary(acp_lreturn)
sd(acp_lreturn)
library(fBasics)
skewness(acp_lreturn)
kurtosis(acp_lreturn)
par(mfrow = c(1,2))
hist(acp_lreturn, xlab = 'daily log returns', breaks = 25, main = 'daily log returns')
boxplot(acp_lreturn)
title('daily log returns')

# question 4
## problem 12 13
data = read.csv("MCD_PriceDaily.csv", head = T)
head(data)
adjPrice = data[, 7]
par(mfrow = c(1,1))
adjp_return = adjPrice[-1]/adjPrice[-1177] - 1
adjp_lreturn = diff(log(adjPrice))
plot(adjp_return, adjp_lreturn, xlab = "Return", ylab = "Log Return")
abline(a = 0,b = 1,col = "red",lwd = 2)
mean(adjp_return)
miu = mean(adjp_lreturn)
sd(adjp_return)
sigma = sd(adjp_lreturn)
adjp_return[1:3]
adjp_lreturn[1:3]

## problem 16 17
niter = 10000 # number of iterations
below = rep(0, niter) # set up storage
set.seed(2015)
for (i in 1:niter)
  {
  r = rnorm(20, mean = miu,
          sd = sigma) # generate random numbers for 20 times
  prices = 93.07 * exp(cumsum(r)) # cumsum means to get the sum of all r for every time point
  indx = (min(prices) < 85) # if minimum price is less than 85 over next 20 days
  below[i] = 100 * indx - (1 - indx)
}
mu = mean(below)
mu + c(-1,1)* sqrt(mu*(1-mu)/niter)*qnorm(0.975)
## 95% CI is larger than 0, we should bet

niter = 10000 # number of iterations
below = rep(0, niter) # set up storage
set.seed(2016)
for (i in 1:niter)
{
  r = rnorm(20, mean = miu,
          sd = sigma) # generate random numbers for 20 times
  prices = 93.07 * exp(cumsum(r)) # cumsum means to get the sum of all r for every time point
  indx1 = (min(prices) < 85) # if minimum price is less than 85 over next 20 days
  indx2 = (min(prices) < 84.5)
  below[i] = 100 * indx1 + 25 * indx2 - (1 - indx1)
}
mu = mean(below)
mu + c(-1,1)* sqrt(mu*(1-mu)/niter)*qnorm(0.975)
```