```{r}
#1
library(rugarch)
rm(list=ls())
load("returns.Jan.19.2024.RData")
lik=function(theta) {
  sigm=matrix(nrow=length(returns[,"AAPL"]),ncol=1)
  a=returns[,"AAPL"]-theta[1]
  sigm[1]=a[1]^2
  for (i in 2:length(returns[,"AAPL"])) {
    sigm[i] = theta[2] + theta[3]*a[i-1]^2 + theta[4]*sigm[i-1]
  }
  lik=-sum(log(ddist(distribution="std",y=returns[,"AAPL"],mu=theta[1],sigma=sqrt(sigm),shape=theta[5])))
  lik
}
lminfit = nlminb(c(mean(returns[,"AAPL"]),0.00001,0.1,0.8,4),lik,lower=c(-1,1e-7,0.000001,0.000002,2.1))
param_vals=lminfit$par
names(param_vals) = c("mu","om","alph","bet","shap")
param_vals
#The mles are listed below in the order of mu,omega,alpha,beta,and nu
```

```
          mu           om         alph          bet         shap
1.361140e-03 9.143004e-06 9.671205e-02 8.793987e-01 4.612894e+00
```

```{r}
gli = ugarchspec(variance.model=list(garchOrder=c(1,1)),mean.model=list(armaOrder=c(0,0)),distribution.model="std")
fit_rugarch=ugarchfit(gli,returns[,"AAPL"])
coef(fit_rugarch)
```

```
          mu        omega       alpha1        beta1        shape
1.361863e-03 9.293764e-06 9.858637e-02 8.774014e-01 4.627523e+00
```

```{r}
fit_rugarch
#The difference between the mles and those from rugarch package is very small, smaller than the standardized error in the table below
```

R Console

data.frame
4 x 3

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(1,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.001362    0.000251   5.4222    0e+00
omega   0.000009    0.000002   4.2677    2e-05
alpha1  0.098586    0.004867  20.2567    0e+00
beta1   0.877401    0.011498  76.3093    0e+00
shape   4.627523    0.317945  14.5545    0e+00

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.001362    0.000251   5.4298 0.000000
omega   0.000009    0.000004   2.2663 0.023433
```

2) (a) we know $\beta_0, \beta_1, x_t, \delta, a_{t-1}$ and hence σ_t

$var(Y_t) = var(a_t) = (0.001 + 0.5\, a_{t-1}^2)\, var(\varepsilon_t)$

$= (0.001 + (0.5)(0.15^2))(1) = 0.01225$

(b) a_t value is not known because you don't know ε_t. Constant times ε_t still follows a normal distribution. So a_t is normal and so Yes Y_t is normal.

(c) a_t is not normal because both noise and σ_t are not known. Somehow it will increase tail weight. As t increases the tail weight will accumulate and will no longer be a normal distribution because now we have 2 randomness sources ε_t and a_{t-1}. In part b a_{t-1} was given so then it was not random. So now Y_t is not normal because it has more tail weight

3) (1) $E\{E(Y|X)\} = \int_{-\infty}^{\infty} E[Y|X] f_X(x)\,dx$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} y\, f_{Y|X}(y|x)\,dy\, f_X(x)\,dx$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} y\, f_{X,Y}(x,y)\,dy\,dx = \int_{-\infty}^{\infty} y \underbrace{\int_{-\infty}^{\infty} f_{X,Y}(x,y)\,dx}_{f_Y(y)}\,dy$$

$$= \int_{-\infty}^{\infty} y\, f_Y(y)\,dy = E(Y)$$

(2) $Var(Y) = E(Y^2) - E(Y)^2$

$E(Y^2) = E(E(Y^2|X)) = E[Var(Y|X) + E(Y|X)^2]$

$E(Y)^2 = E(E(Y|X))^2$

$E(Var(Y|X)) + \underbrace{E(E(Y|X)^2) - E(E(Y|X))^2}_{Var(E(Y|X))}$

$= E(Var(Y|X)) + Var(E(Y|X))$

```
```{r}
#4a
rm(list=ls())
load("homework7.RData")
param1_fit=lm(aapl~market -1)
summary(param1_fit)

#based on the table below beta for model 1 is 1.1381
```
```

```

Call:
lm(formula = aapl ~ market - 1)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.7011 -0.5435 -0.0154  0.5789  7.0812 

Coefficients:
       Estimate Std. Error t value Pr(>|t|)    
market   1.1381     0.0425   26.78   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.174 on 999 degrees of freedom
Multiple R-squared:  0.4178,	Adjusted R-squared:  0.4173 
F-statistic:   717 on 1 and 999 DF,  p-value: < 2.2e-16
```

```{r}
new_mod = cbind(c(market[1:250],rep(0,750)),c(rep(0,250),market[251:500],rep(0,500)),c(rep(0,500),market[501:750],rep(0,250)),c(rep(0,750),market[751:1000]))
param_new_fit = lm(aapl~new_mod-1)
summary(param_new_fit)
#the four estimates betas are 1.13963,0.92955,1.25553 and 1.24092 as per the table below.
```

```
Call:
lm(formula = aapl ~ new_mod - 1)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.6573 -0.5434 -0.0398  0.5681  7.0834 

Coefficients:
         Estimate Std. Error t value Pr(>|t|)    
new_mod1  1.13963    0.07619  14.958  < 2e-16 ***
new_mod2  0.92955    0.08782  10.585  < 2e-16 ***
new_mod3  1.25553    0.16058   7.819 1.35e-14 ***
new_mod4  1.24092    0.06806  18.233  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.171 on 996 degrees of freedom
Multiple R-squared:  0.4227,	Adjusted R-squared:  0.4204 
F-statistic: 182.3 on 4 and 996 DF,  p-value: < 2.2e-16
```

```
```{r}
#4b
anova(param1_fit,param_new_fit)
#p value is 0.03798 which is less than 0.05 so we reject the null hypothesis and go with the second model.
```
```

```
Analysis of Variance Table

Model 1: aapl ~ market - 1
Model 2: aapl ~ new_mod - 1
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1    999 1376.6
2    996 1365.0  3    11.591 2.8192 0.03798 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```