FTAP Homework 9

Zachary Fogelson

July 23, 2015

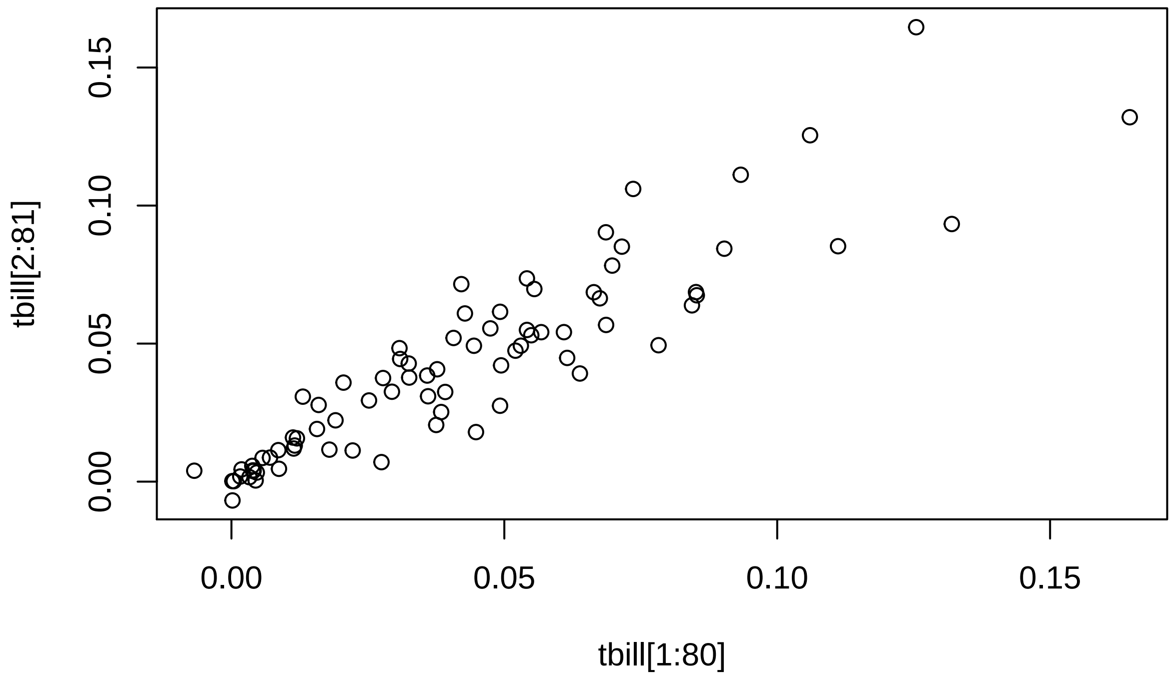
Reading

```
gdp <- read.xls("gdp.xls", skip = 17)

gdpGrowth <-

tb <- read.csv("tbill.csv")

sp500 <- read.csv("table.csv")
sp500$DateTime <- as.Date(sp500$Date, "%Y-%m-%d")
sp500 <- sp500[order(sp500$DateTime, decreasing=F),]
sp500 <- sp500[complete.cases(sp500),]

hp <- read.csv("table (1).csv")
hp$DateTime <- as.Date(hp$Date, "%Y-%m-%d")
hp <- hp[order(hp$DateTime, decreasing=F),]
hp <- hp[complete.cases(hp),]

spRet <- diff(log(sp500$Adj.Close))
hpRet <- diff(log(hp$Adj.Close))
```

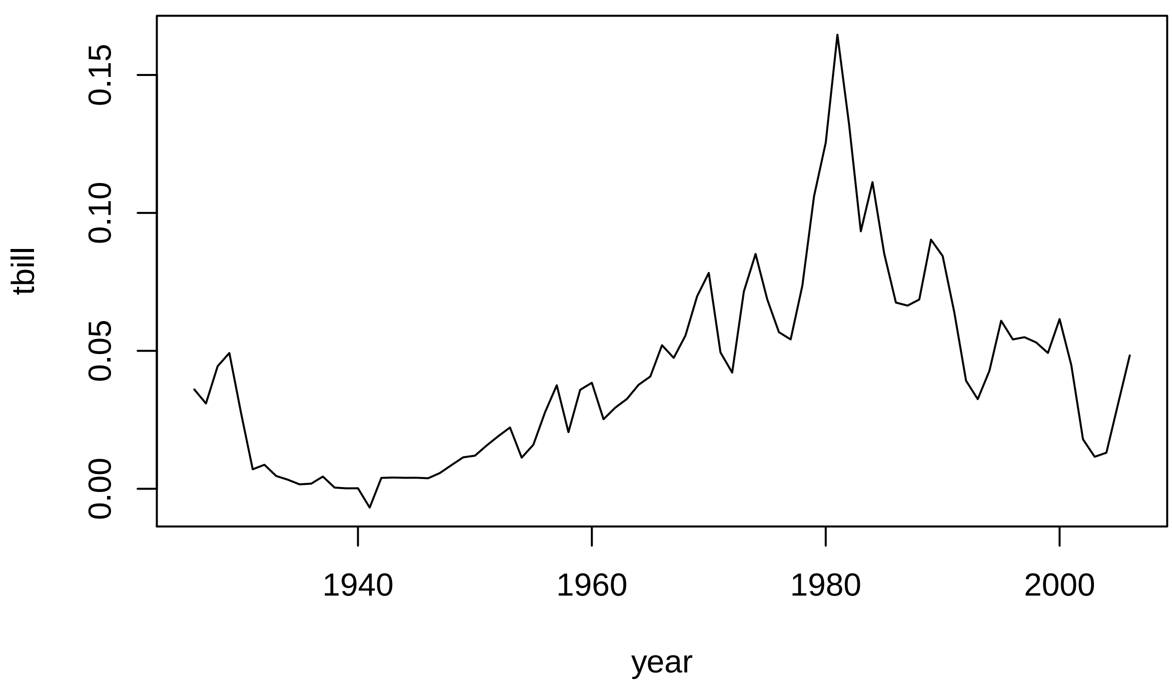
Problem 1

Set up

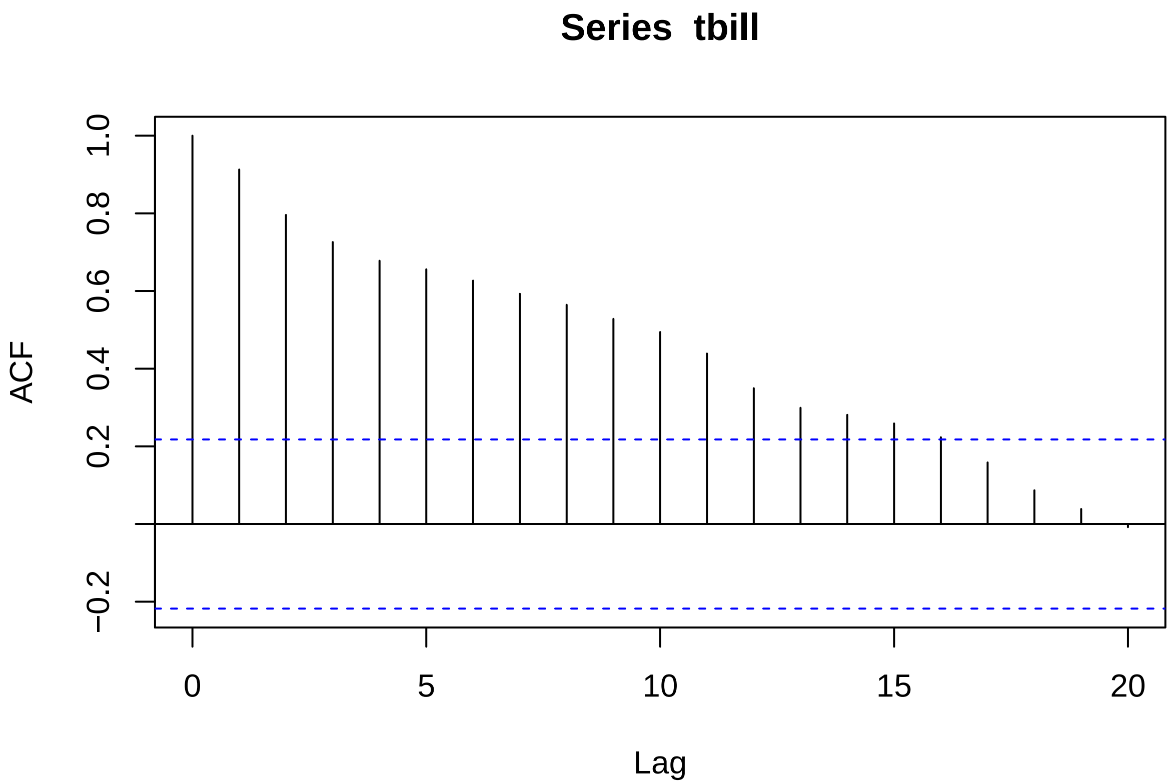
```
year<-tb$Year
tbill<-tb$tbill
plot(tbill[1:80], tbill[2:81])
```



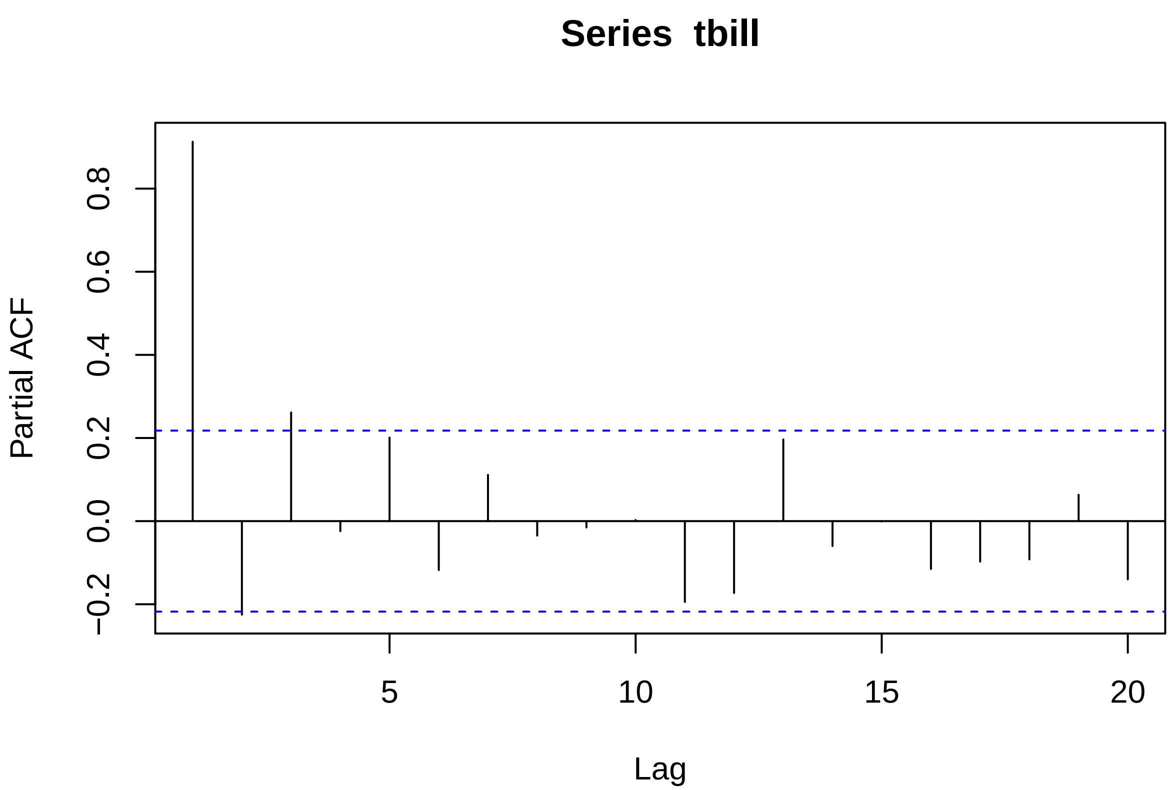

```
plot(year,tbill, type="l")
```




```
acf(tbill,lag.max=20)
```




```r
pacf(tbill,lag.max=20)
```



```r
ar<-arima(tbill,order=c(1,0,0))
summary(ar)
```

```
##
## Call:
## arima(x = tbill, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.9037     0.0417
## s.e.  0.0430     0.0145
##
## sigma^2 estimated as 0.0001948:  log likelihood = 230.22,  aic = -454.45
##
## Training set error measures:
##                        ME       RMSE       MAE       MPE    MAPE      MASE
## Training set 9.692555e-05 0.01395855 0.0103087 -97.49674 121.474 0.9790536
##                   ACF1
## Training set 0.2191448
```

a

```r
p<-predict(ar,n.ahead=15)
kable(head(data.frame(Predictions = c(p$pred))))
```

Predictions

0.0477050 0.0471221 0.0465954 0.0461194 0.0456892 0.0453005

b As we predict further out, our prediction for our errors changes over time:

Lets find the forecast error variance:

$$Var\left(e_t^k\right)=Var\left(\sum_{j=0}^{k-1}\beta_1^j\varepsilon_{t+k-j}\right)$$

but the ε_t are iid. So…

$$Var\left(e_t^k\right)=\sum_{j=0}^{k-1}\beta_1^{2j}Var\left(\varepsilon_{t+k-j}\right)=\sigma^2\sum_{j=0}^{k-1}\beta_1^{2j}=\frac{\left(1-\beta_1^{2k}\right)}{\left(1-\beta_1^2\right)}\sigma^2$$

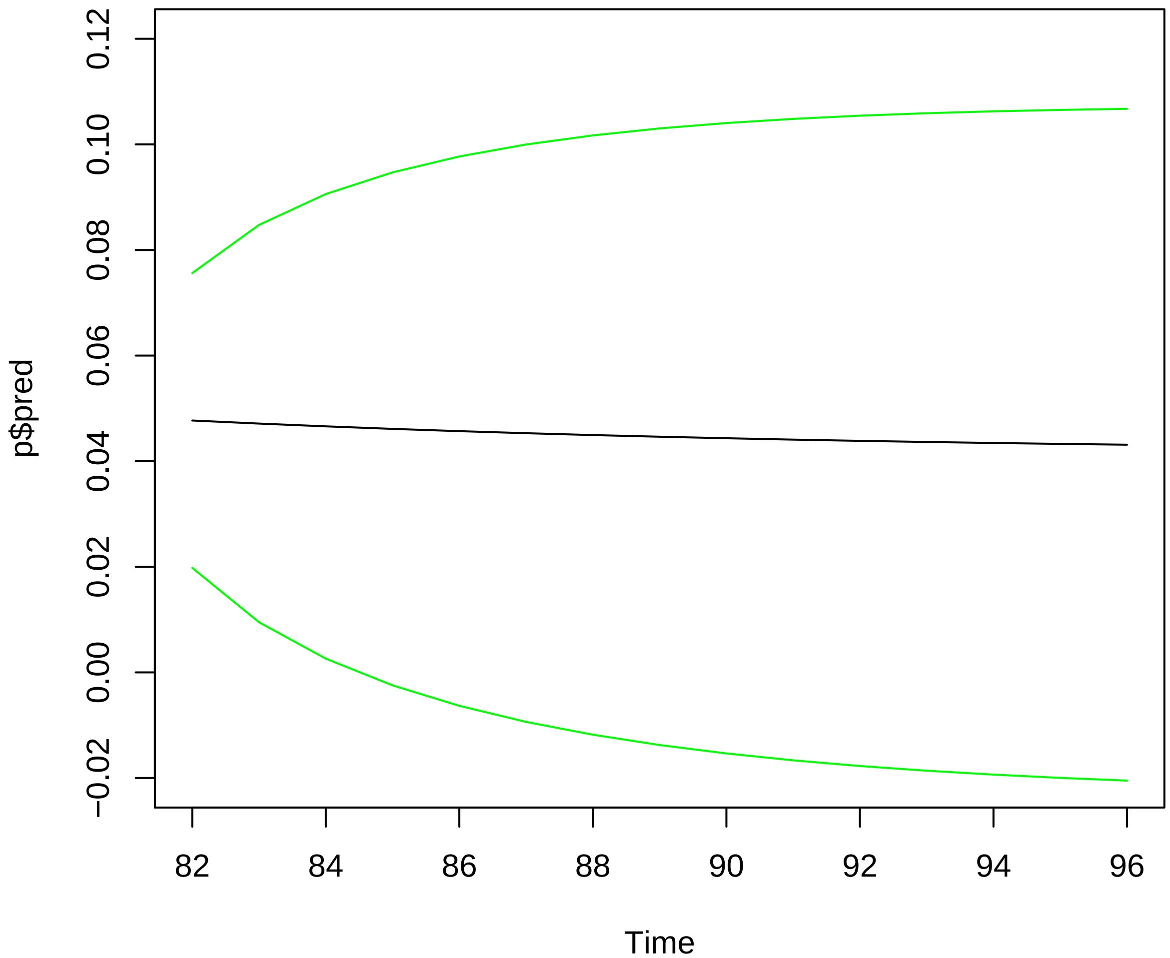
```r
p<-predict(ar,n.ahead=15)
kable(head(data.frame(Variance = c(p$se^2))))
```

Variance

0.0001948 0.0003540 0.0004839 0.0005900 0.0006767 0.0007474

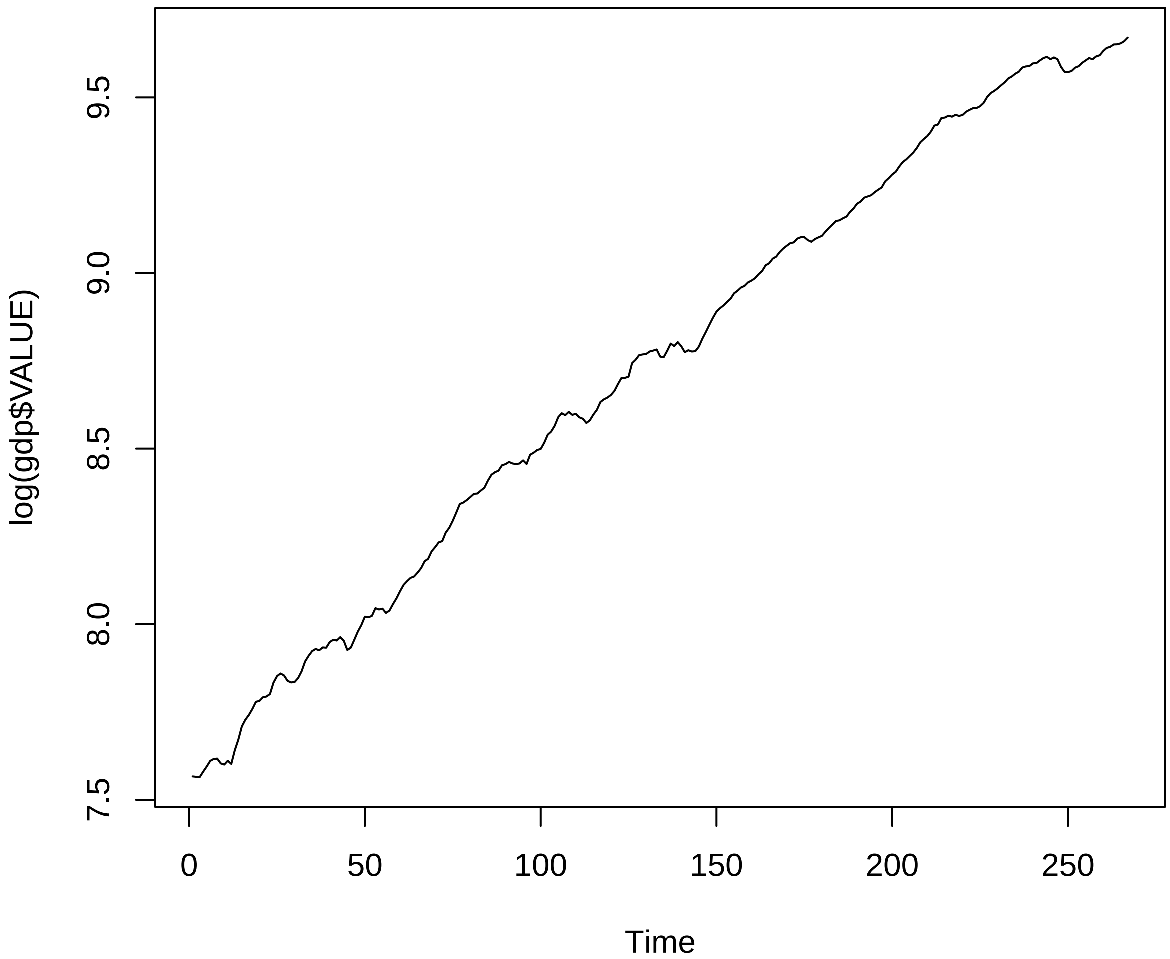
c

```r
plot(p$pred, ylim = c(-.02,.12))
lines(p$pred + 2 * p$se, col = "green")
lines(p$pred - 2 * p$se, col = "green")
```



Problem 2

```r
plot.ts(log(gdp$VALUE))
```



a

```
grwRate <- diff(log(gdp$VALUE))
mean(grwRate)
```

```
## [1] 0.007908376
```

```
rwfPred <- rwf(log(gdp$VALUE), h=4, drift=T, level=c(80,95), fan=FALSE, lambda=NULL)
rwfPred$model$drift
```

```
## [1] 0.007908376
```

```
kable(head(data.frame(Upper = c(as.numeric(exp(rwfPred$upper[,2]))))))
```

Upper

16270.24 16529.38 16762.41 16982.96

```
kable(head(data.frame(Upper = c(as.numeric(exp(rwfPred$lower[,2]))))))
```

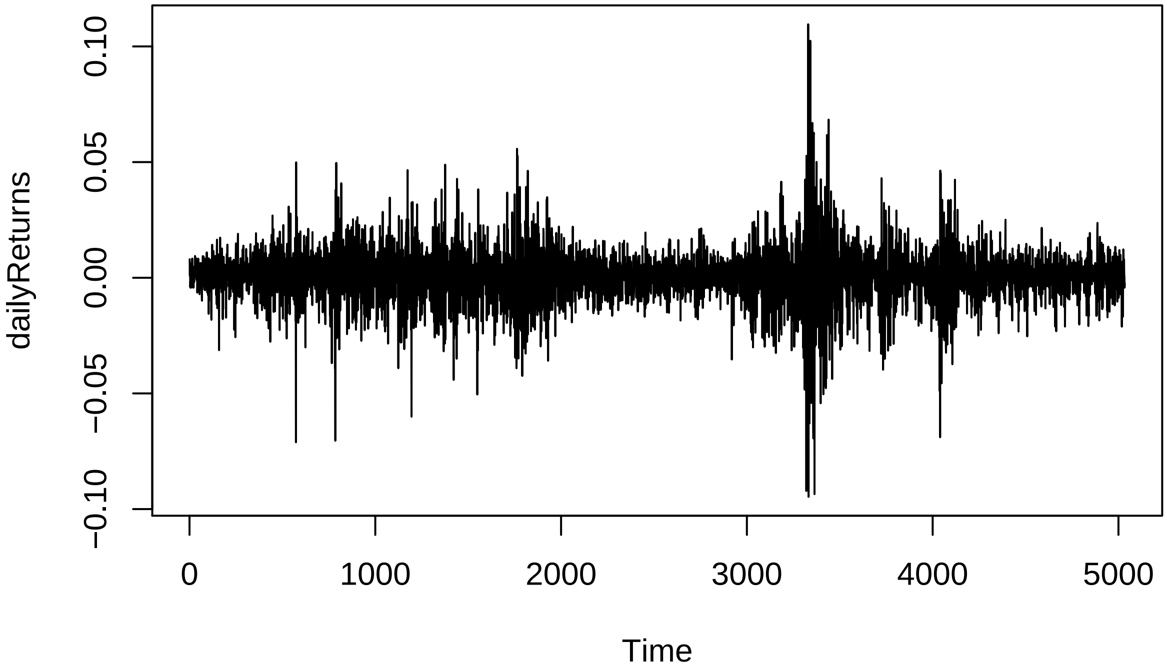
Upper

15665.60 15665.84 15694.34 15737.48

Problem 3

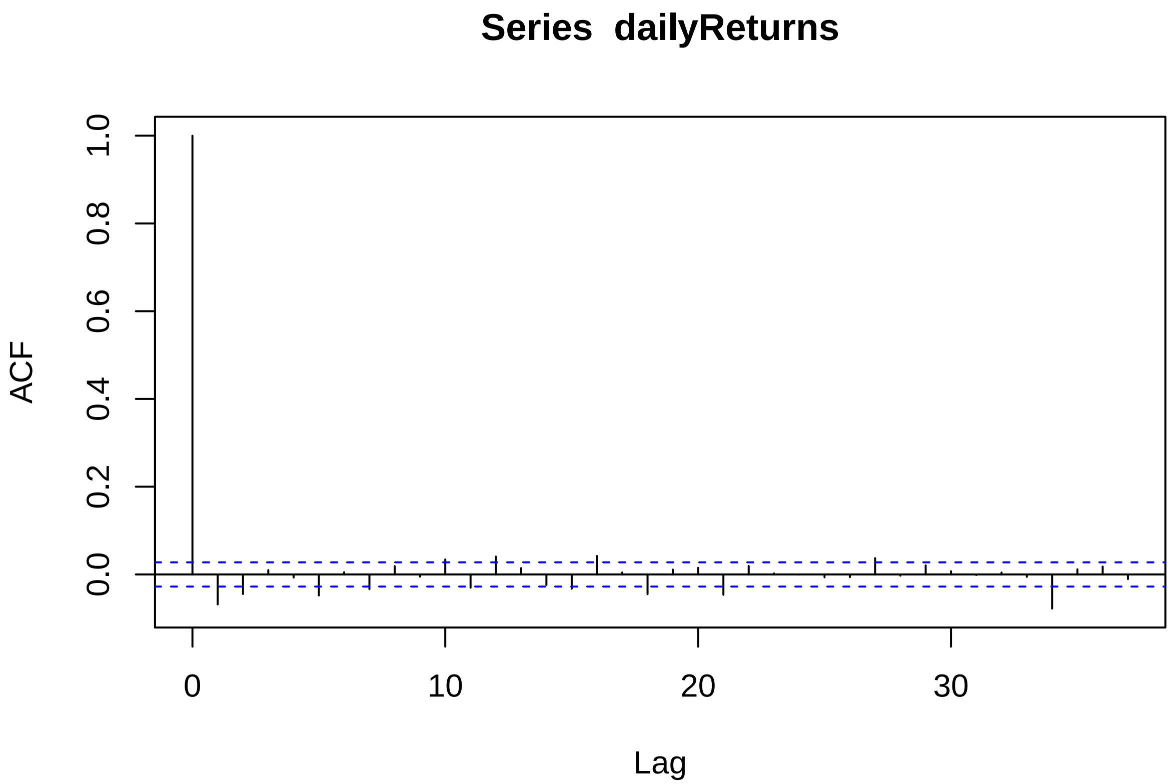
a

```
spRet <- diff(log(sp500$Adj.Close))  # This should have already been defined but R doesn't think so
dailyReturns <- spRet
plot.ts(dailyReturns)
```



b

```
acf(dailyReturns)
```



Because the auto-correlations are consistently higher than the critical value, we reject the null hypothesis that the difference in absolute daily returns is not auto-correlated. Namely, we concluded that there is a relationship between the absolute value of yesterday's returns and today's returns.

c

```
kurtosis(dailyReturns)[1]
```

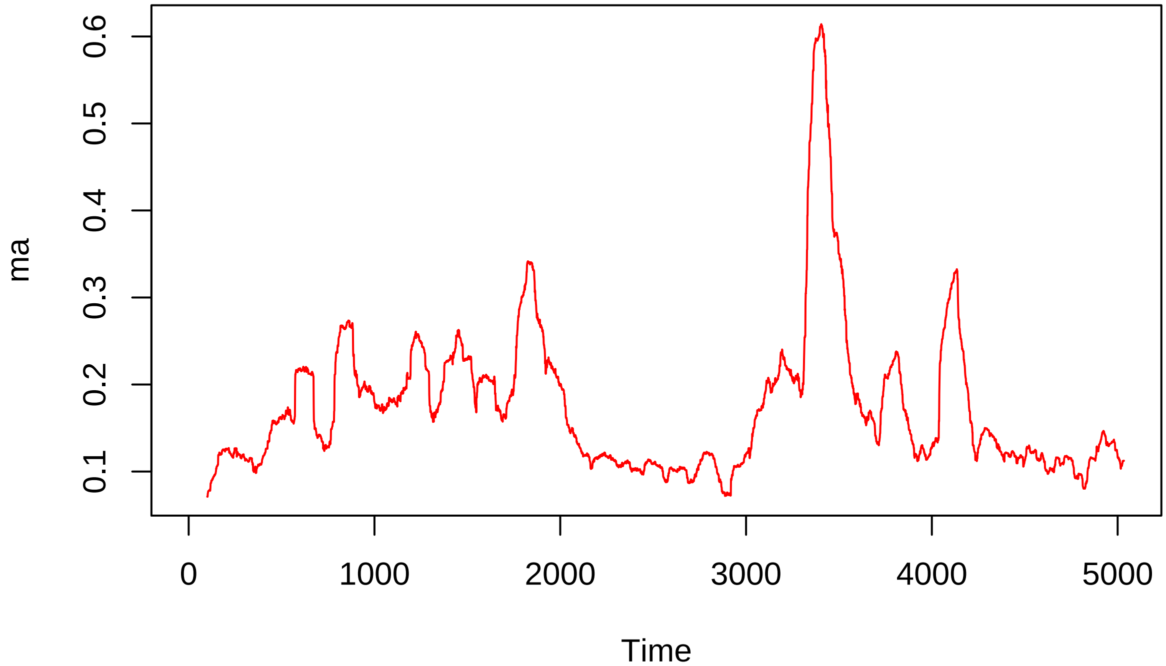
```
## [1] 7.983142
```

R returns the excess Kurtosis over and above a normal distrobution. Therefore, we can see that the returns have MUCH fatter tails than a normal distrobution would suggest.

Problem 4

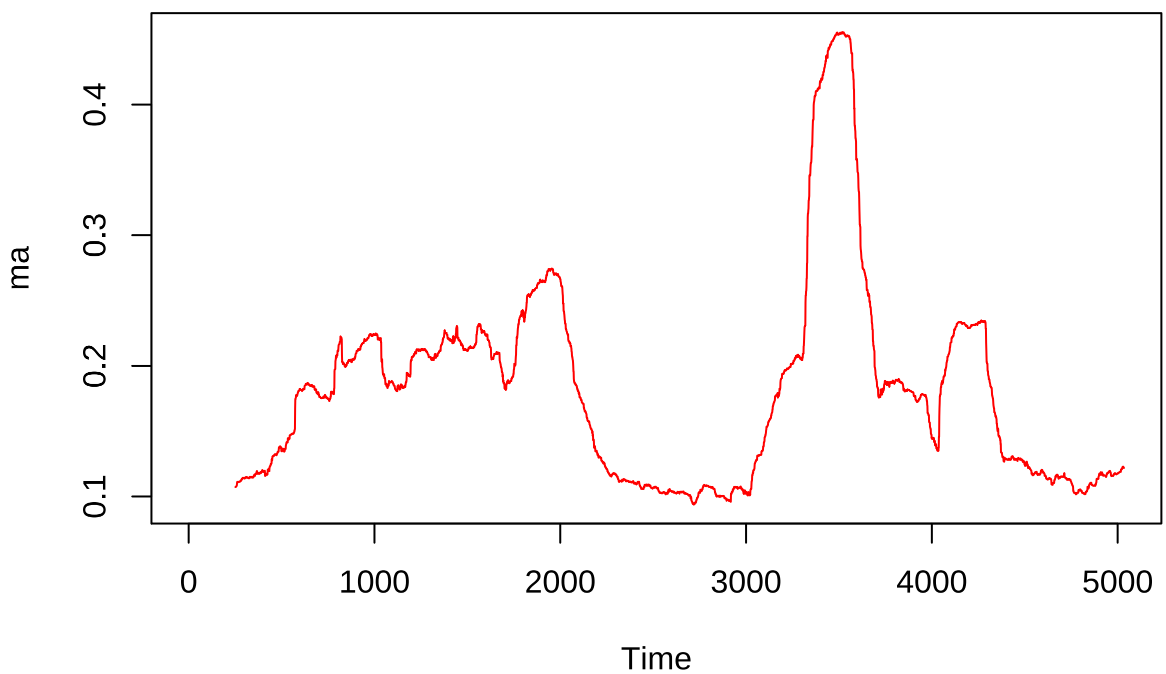
a

```
r2=252*dailyReturns*dailyReturns
ar=sqrt(252)*r2
ma <- sqrt(filter(r2,rep(1/100,100), sides=1))
plot(ma,type="l",col="red")
```



b

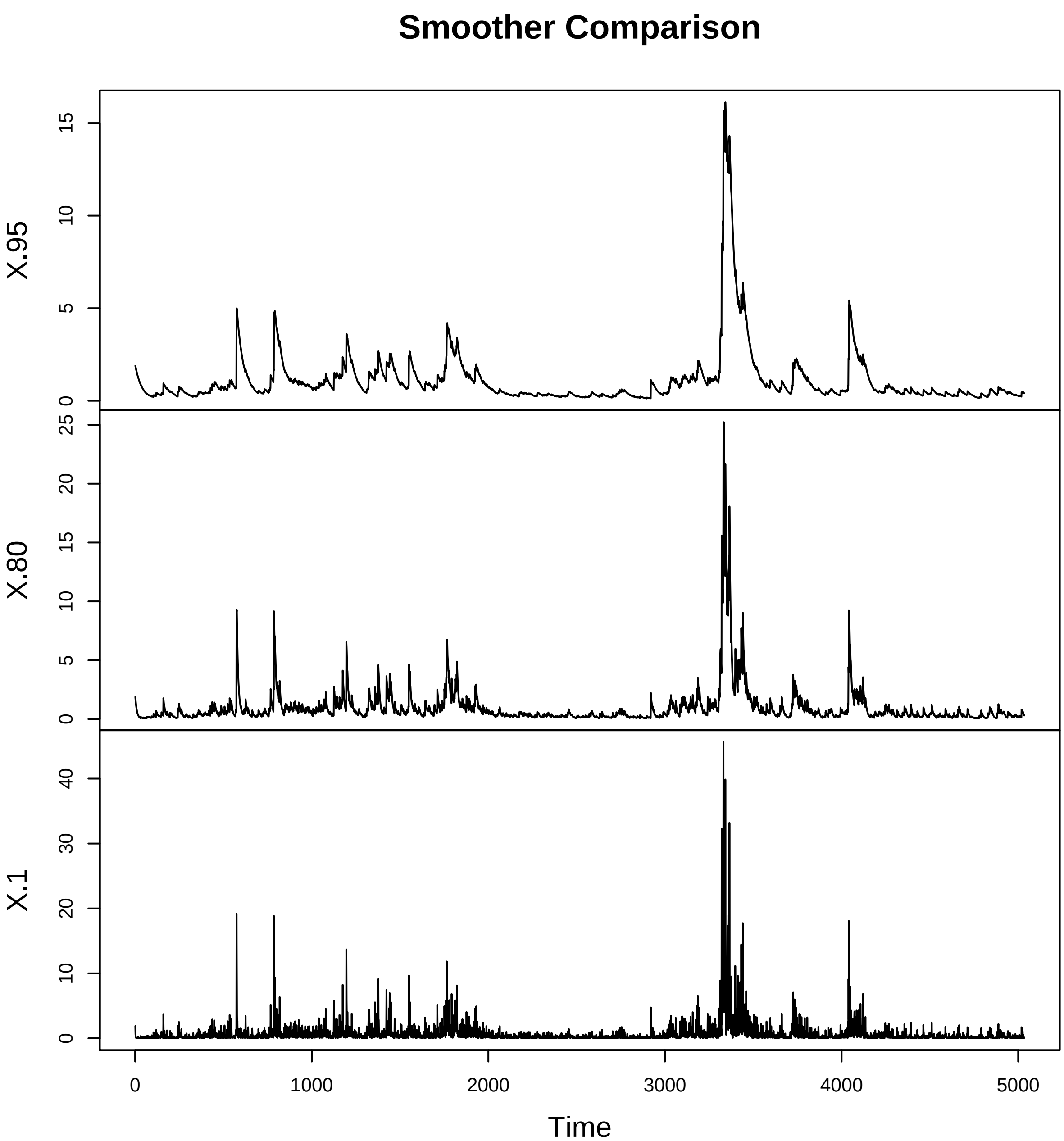
```r
ma <- sqrt(filter(r2,rep(1/252,252), sides=1))
plot(ma,type="l",col="red")
```



Problem 5

a, b, c

```
smoother <- function(lambda){
  rmvar <- matrix(0,nrow=length(ar),ncol=1)
  rmvar[1,1] <- var(ar)
  for (i in 2:length(rmvar)){
    rmvar[i,1] <- (lambda*rmvar[i-1,1])+((01-lambda)*ar[i-1]^2)
  }
  sqrt(rmvar)
}

plot.ts(data.frame(".95" = c(smoother(.95)), ".80" = c(smoother(.8)), ".1" = c(smoother(.1))), main = "S
```



Problem 6

a

```
myGarch <- function(r, h){
  .00045 + .04^2*r^2 + .94*h
}
myGarch(.04, .03)
```

```
## [1] 0.02865256
```

b

```
myGarch(.01, .03)
```

```
## [1] 0.02865016
```

c

$$\sigma^2 = \frac{\omega}{1-\alpha-\beta}$$

```
.00045/(1-.04-.94)
```

```
## [1] 0.0225
```

Problem 7

a

```
a1 <- garchFit(~garch(1,0),data=dailyReturns, trace = F)
-1*a1@fit$llh
```

```
## LogLikelihood
##      15249.14
```

b

```
a9 <- garchFit(~garch(9,0),data=dailyReturns, trace = F)
-1*a9@fit$llh
```

```
## LogLikelihood
##       16041.5
```

c

```r
g11 <- garchFit(~garch(1,1),data=dailyReturns, trace = F)
-1*g11@fit$llh
```

```
## LogLikelihood
##      16051.91
```

d

```r
g22 <- garchFit(~garch(2,2),data=dailyReturns, trace = F)
-1*g22@fit$llh
```

```
## LogLikelihood
##      16070.48
```

e Based on the log likelihood values, my garch22 is performing the best.

Problem 8

a

```r
dailyVar <- volatility(g22)^2
```

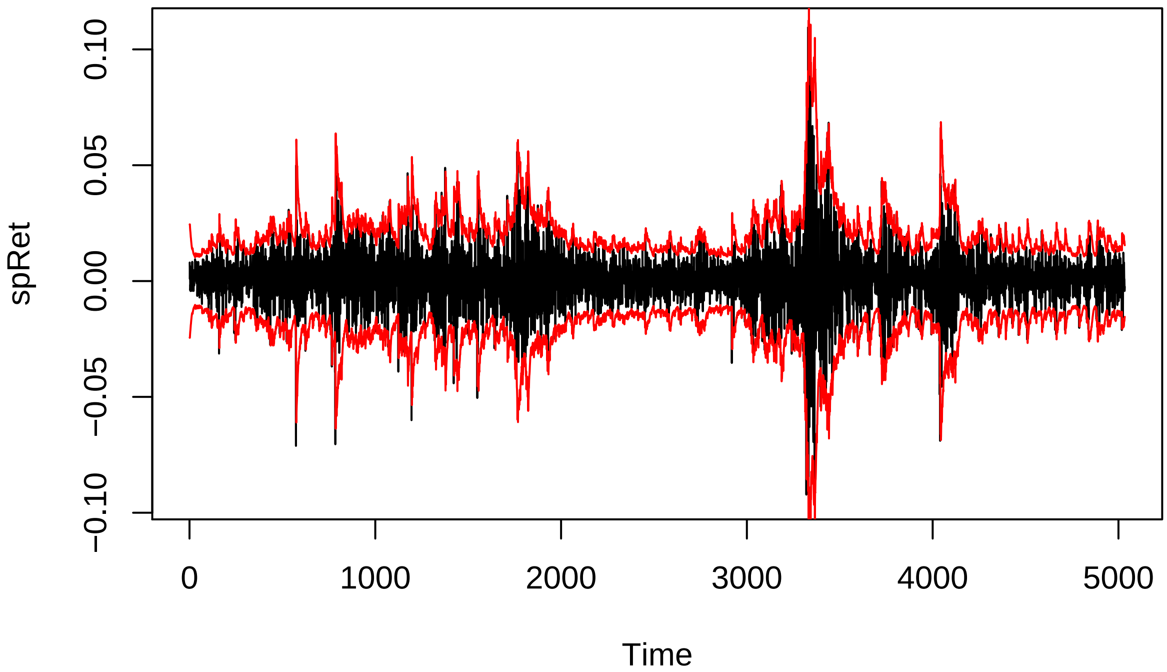
b

```r
interval <- data.frame(Upper <- c(sqrt(g22@h.t)*2), Lower <- c(-sqrt(g22@h.t)*2))

100 * (sum(as.numeric(sp500$Adj.Close[-1] < interval$Upper & sp500$Adj.Close[-1]) > interval$Lower)/leng
```

```
## [1] 99.98014
```

c

```r
plot.ts(spRet)
lines(interval$Upper, col = "red")
lines(interval$Lower, col = "red")
```



Problem 9

a

```r
diffHp<- hpRet
diffSp <- spRet
skewness(diffHp)
```

```
## [1] -0.2303797
## attr(,"method")
## [1] "moment"
```

```r
kurtosis(diffHp)
```

```
## [1] 6.733811
## attr(,"method")
## [1] "excess"
```

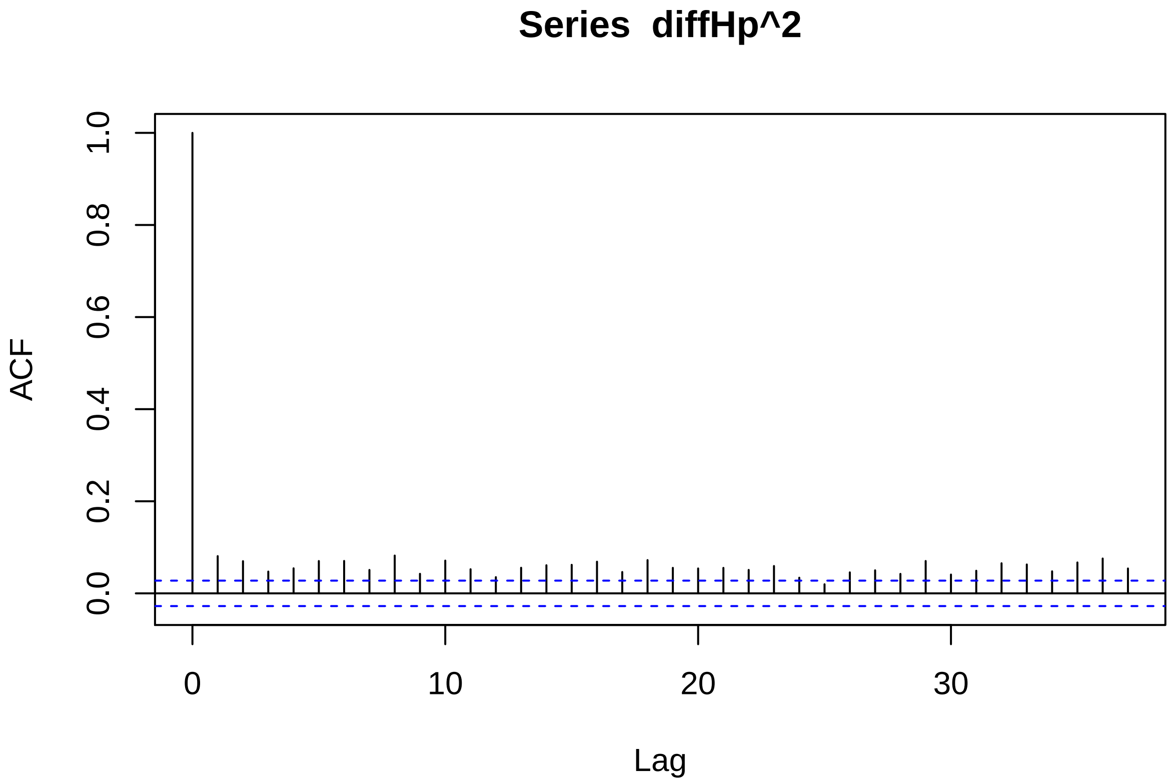
```r
skewness(diffSp)
```

```
## [1] -0.2392782
## attr(,"method")
## [1] "moment"
```

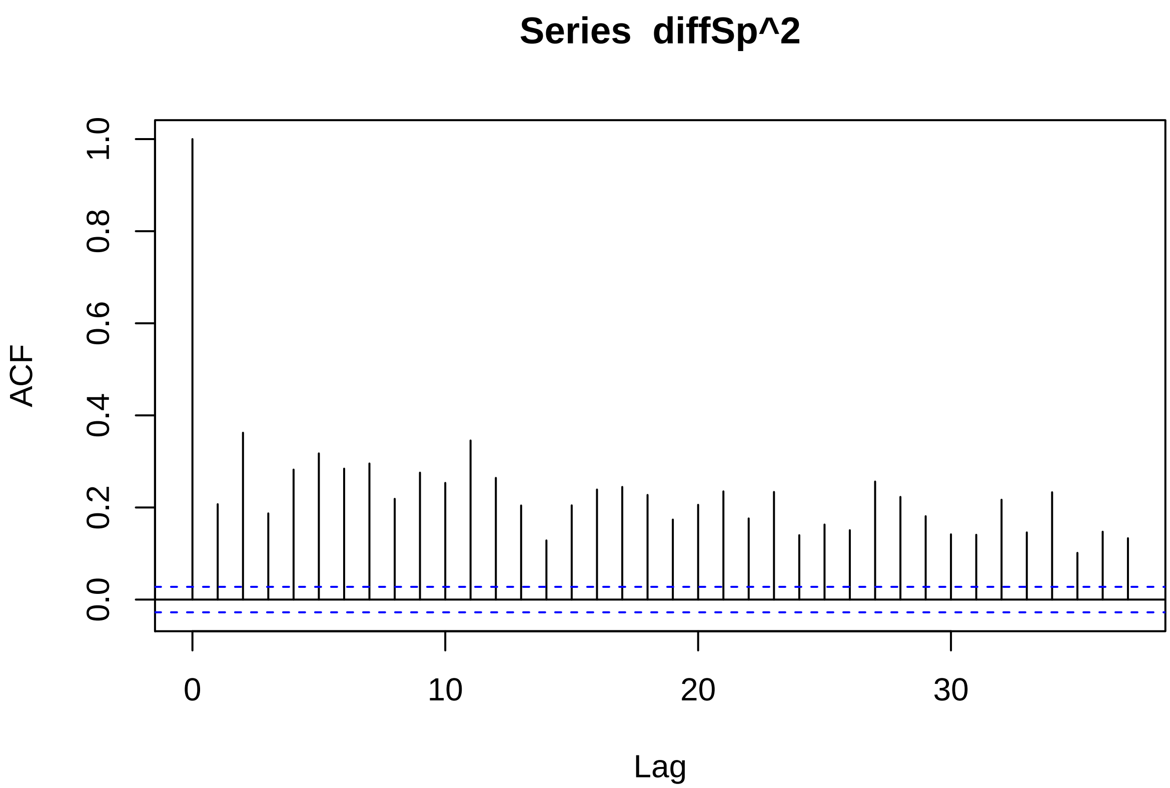
```r
kurtosis(diffSp)
```

```
## [1] 7.983142
## attr(,"method")
## [1] "excess"
```

```r
acf(diffHp^2)
```




```r
acf(diffSp^2)
```



b

```r
hpGarch <- garchFit(~garch(1,1),data=diffHp, trace = F)
summary(hpGarch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = diffHp, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fc98d1db188>
##  [data = diffHp]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 5.6091e-04  3.3701e-06  2.6881e-02  9.6793e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     5.609e-04   2.997e-04    1.872   0.0612 .
## omega  3.370e-06   6.379e-07    5.283 1.27e-07 ***
## alpha1 2.688e-02   2.944e-03    9.132  < 2e-16 ***
## beta1  9.679e-01   3.398e-03  284.869  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  11811.06    normalized:  2.345791
##
## Description:
##  Fri Jul 24 03:05:37 2015 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  14714.79  0
##  Shapiro-Wilk Test  R    W      NA        NA
##  Ljung-Box Test     R    Q(10)  11.12148  0.348132
##  Ljung-Box Test     R    Q(15)  18.87654  0.2193833
##  Ljung-Box Test     R    Q(20)  26.03588  0.1646303
##  Ljung-Box Test     R^2  Q(10)  2.440057  0.9917207
##  Ljung-Box Test     R^2  Q(15)  5.209898  0.9901951
##  Ljung-Box Test     R^2  Q(20)  6.025468  0.9988627
##  LM Arch Test       R    TR^2   3.982232  0.9837549
```

```
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -4.689993 -4.684810 -4.689994 -4.688177
```

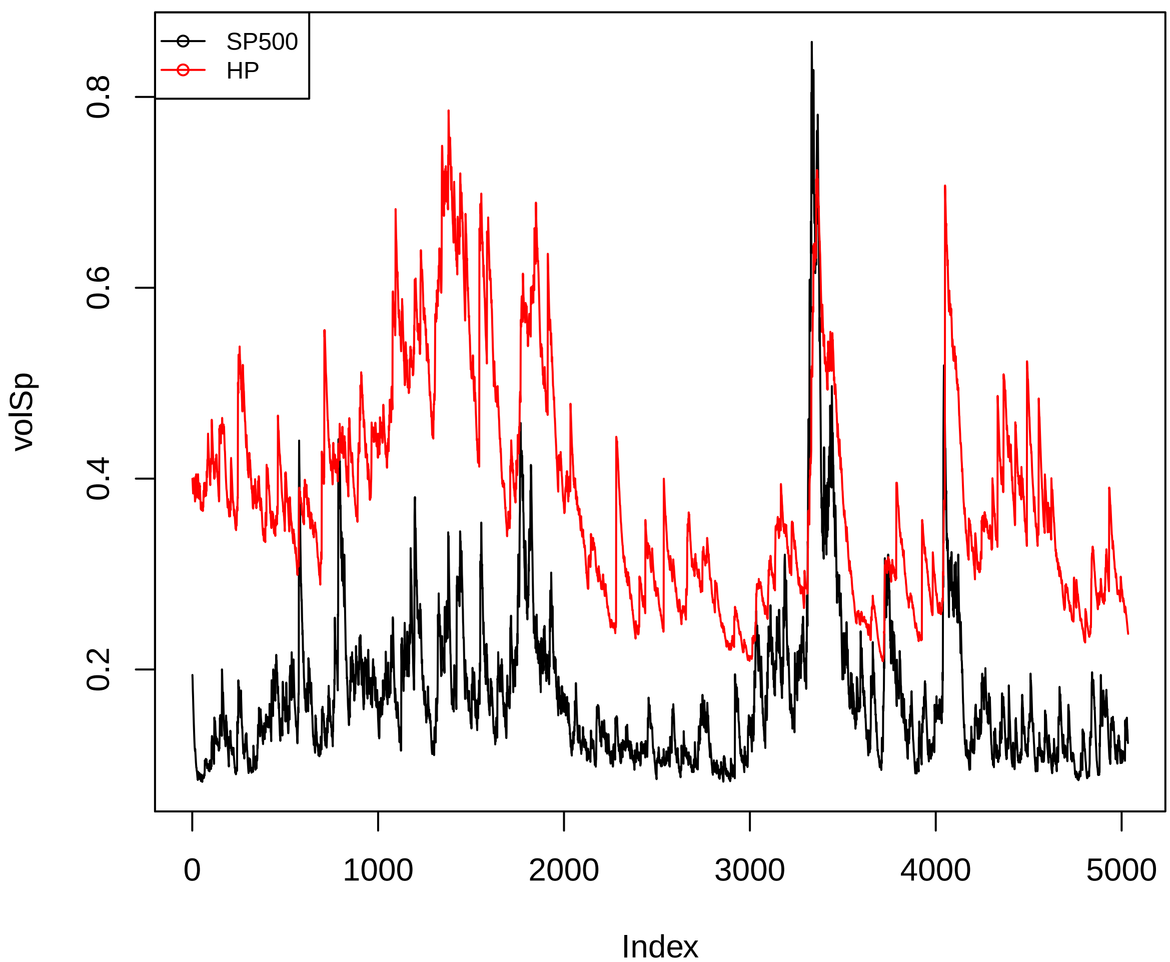
```
volHp <- sqrt(252)*volatility(hpGarch)
```

```
spGarch <- garchFit(~garch(1,1),data=diffSp, trace = F)
summary(spGarch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = diffSp, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fc98eb71c70>
##  [data = diffSp]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 5.8442e-04  1.7353e-06  9.5242e-02  8.9314e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     5.844e-04   1.201e-04    4.867 1.13e-06 ***
## omega  1.735e-06   2.987e-07    5.809 6.27e-09 ***
## alpha1 9.524e-02   8.446e-03   11.277  < 2e-16 ***
## beta1  8.931e-01   9.022e-03   98.994  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  16051.91    normalized:  3.188699
##
## Description:
##  Fri Jul 24 03:05:37 2015 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  631.7441  0
##  Shapiro-Wilk Test  R    W      NA        NA
##  Ljung-Box Test     R    Q(10)  22.44179  0.01300552
```

```
##  Ljung-Box Test     R    Q(15)  30.15053  0.01139014
##  Ljung-Box Test     R    Q(20)  33.9854   0.02622361
##  Ljung-Box Test     R^2  Q(10)  19.87282  0.03047884
##  Ljung-Box Test     R^2  Q(15)  24.46184  0.05765436
##  Ljung-Box Test     R^2  Q(20)  25.35356  0.188238
##  LM Arch Test       R    TR^2   19.08862  0.08641206
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -6.375809 -6.370625 -6.375810 -6.373993
```

```
volSp <- sqrt(252)*volatility(spGarch)
plot(volSp, type = "l")
lines(volHp, col = "red")
legend("topleft", c("SP500","HP"), pch=1, col=c('black', 'red'), lty=1, cex=.75)
```



The annualized volatility of the S&P500 appears to have less volatility over all but there is one voltile day that exceeds all of the volatility of HP.

c

```
as.numeric(spGarch@fit$coef[2])/(1-as.numeric(spGarch@fit$coef[3])-as.numeric(spGarch@fit$coef[4]))
```

```
## [1] 0.0001494093
```

```r
var(diffSp)
```

```
## [1] 0.0001496828
```

```r
as.numeric(hpGarch@fit$coef[2])/(1-as.numeric(hpGarch@fit$coef[3])-as.numeric(hpGarch@fit$coef[4]))
```

```
## [1] 0.0006498927
```

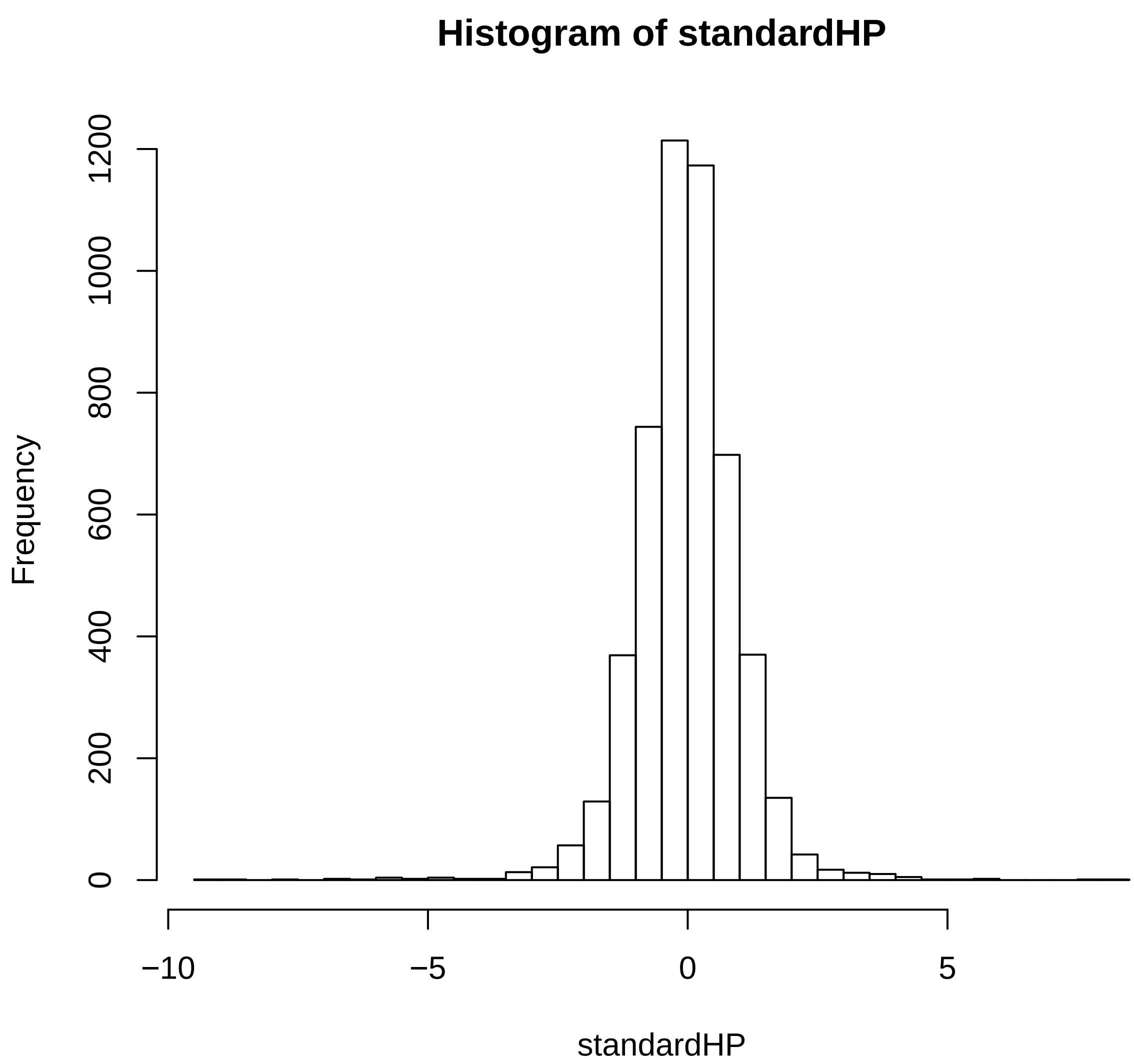
```r
var(diffHp)
```

```
## [1] 0.0006345835
```

The values are extremely close!

d

```r
standardHP <- residuals(hpGarch, standardize = T)
standardSP <- residuals(spGarch, standardize = T)

hist(standardHP, breaks = 30)
```



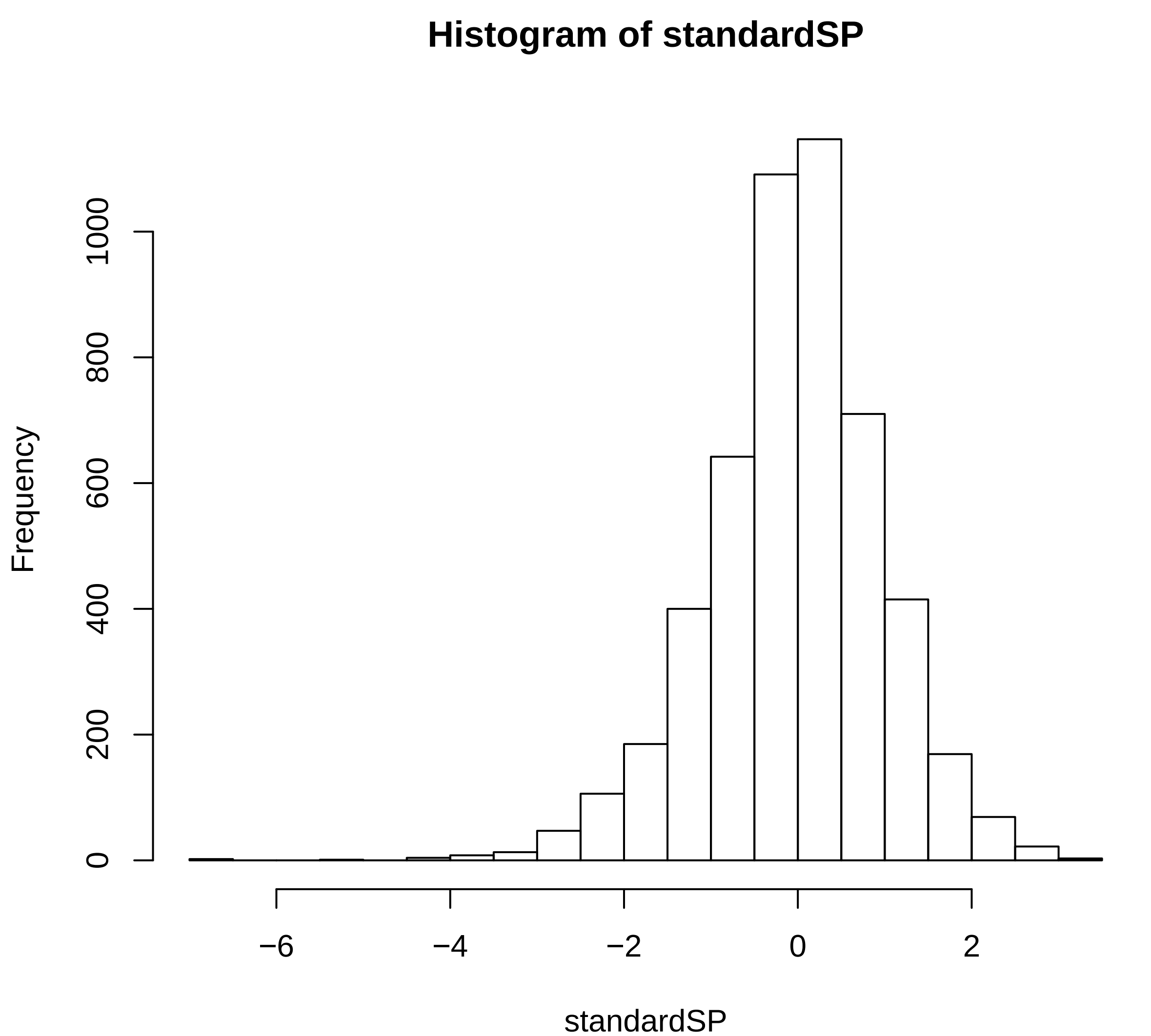
```r
skewness(standardHP)
```

```
## [1] -0.3918477
## attr(,"method")
## [1] "moment"
```

```r
kurtosis(standardHP)
```

```
## [1] 8.33369
## attr(,"method")
## [1] "excess"
```

```r
hist(standardSP, breaks = 30)
```




```r
skewness(standardSP)
```

```
## [1] -0.4457658
## attr(,"method")
## [1] "moment"
```

```r
kurtosis(standardSP)
```

```
## [1] 1.487038
## attr(,"method")
## [1] "excess"
```

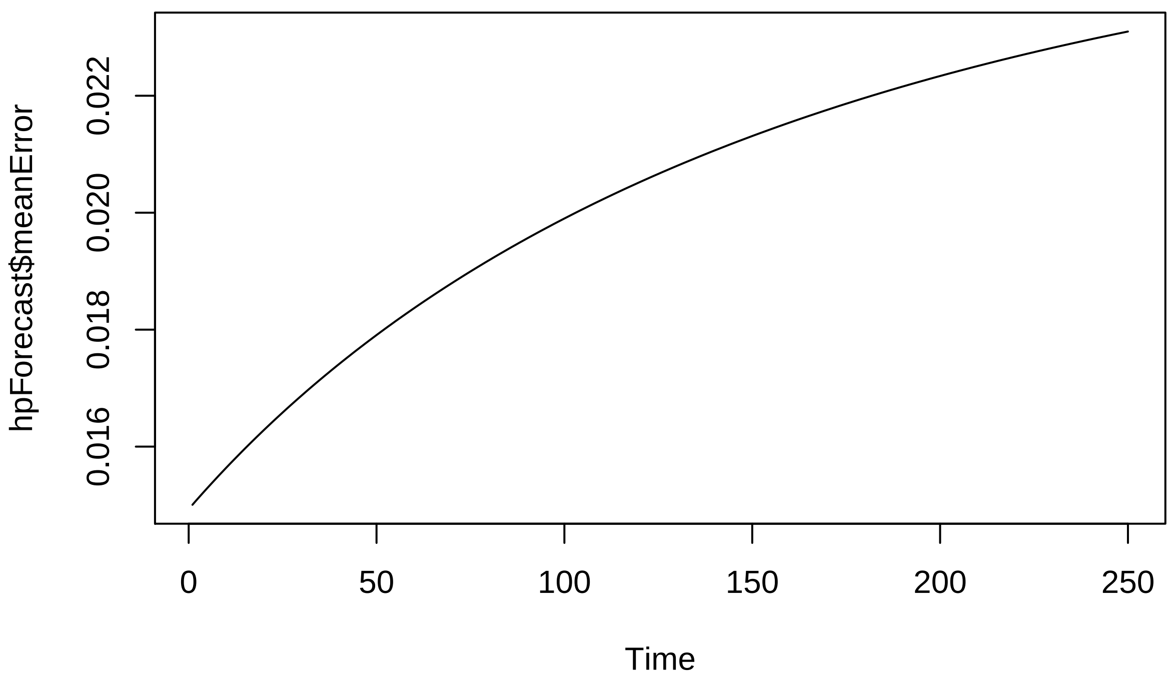
My histogram looks normal but the kortousis is far in excess of normal.

e

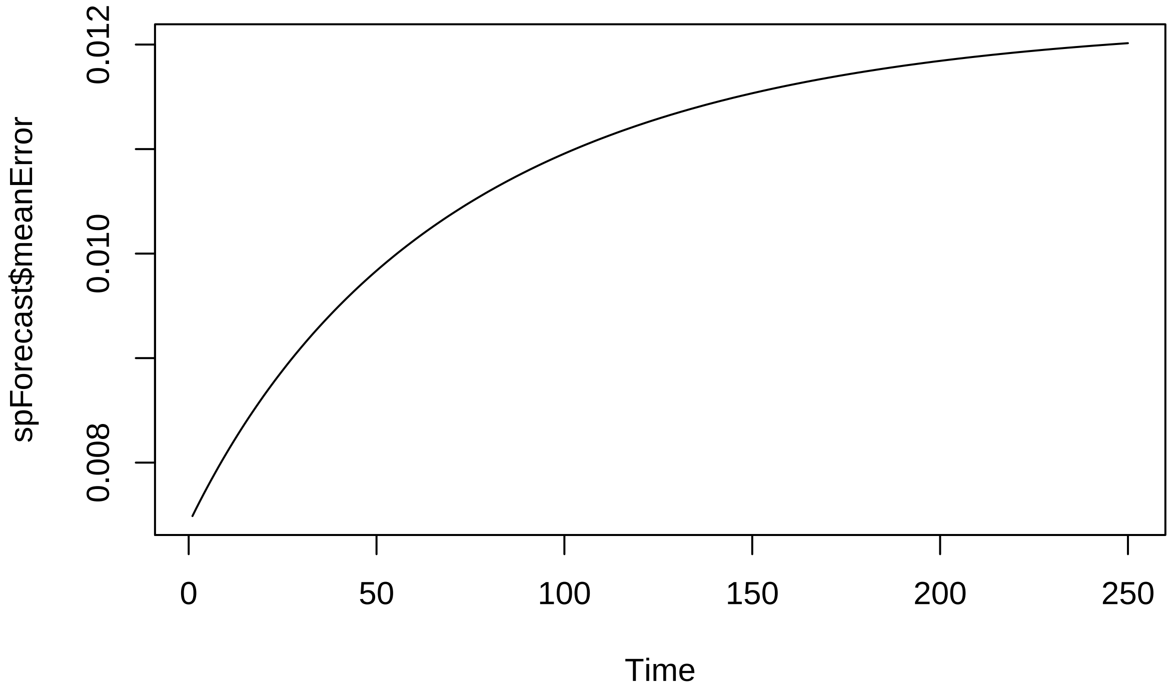
Based on the statistics, the the tails are much fatter than a normal distrobution so you would fail to predict outliers.

f

```r
hpForecast<-predict(hpGarch,n.ahead=250)
ts.plot(hpForecast$meanError,type="l")
```




```r
spForecast<-predict(spGarch,n.ahead=250)
ts.plot(spForecast$meanError,type="l")
```



g

```
mean(hpForecast$standardDeviation^2)
```

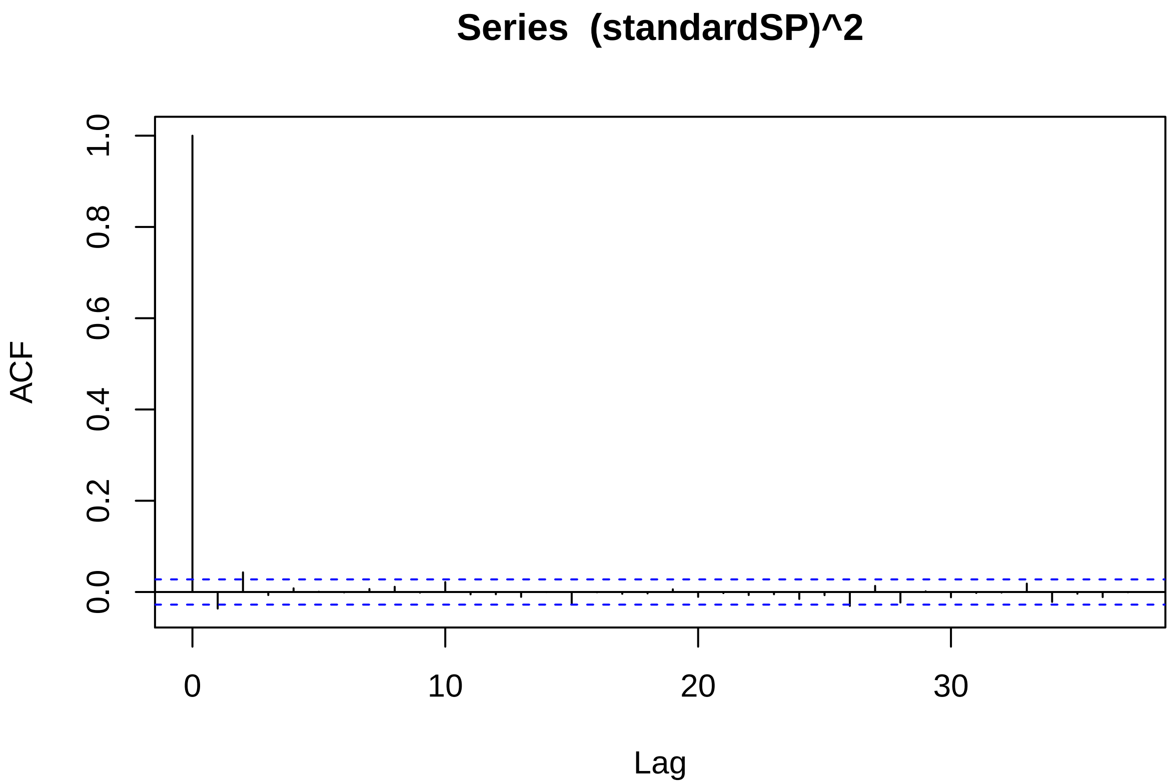
```
## [1] 0.0004115751
```

```
mean(spForecast$standardDeviation^2)
```

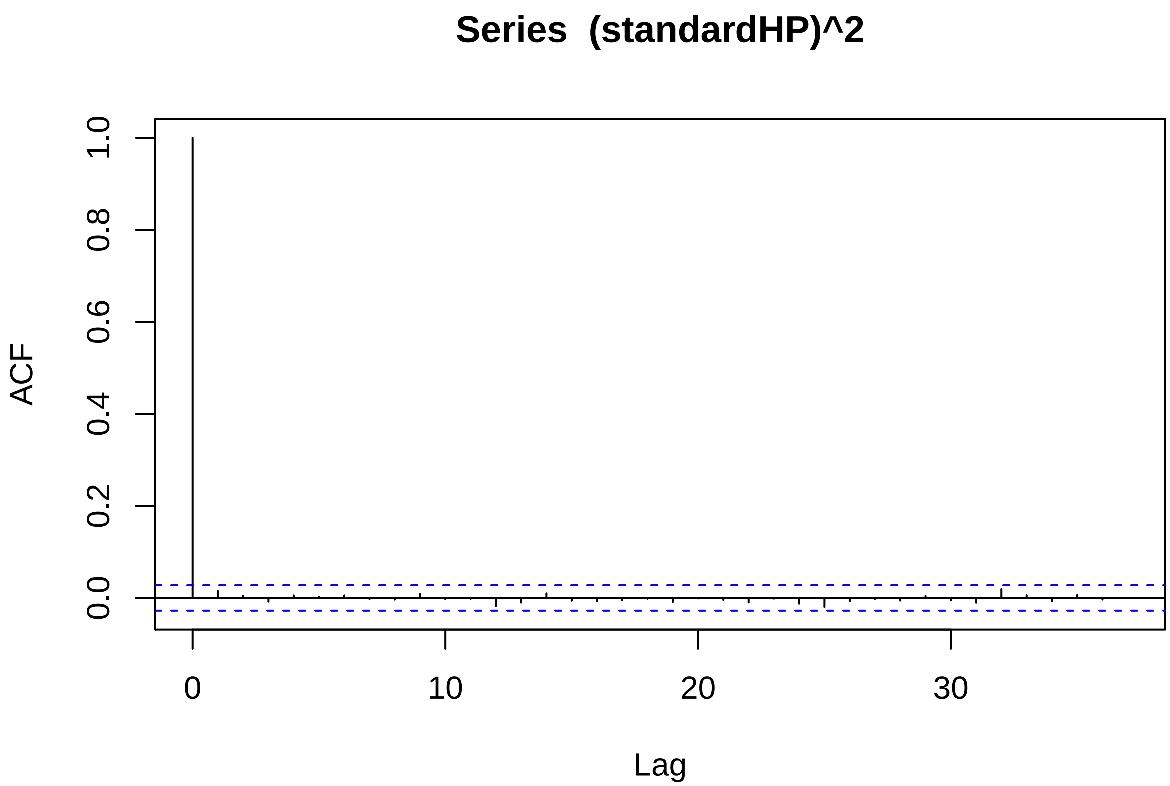
```
## [1] 0.0001189986
```

h

```
acf((standardSP)^2)
```




```
acf((standardHP)^2)
```



Problem 10

a

The mean of the profile should be the expectation of the linear combination of the weighted portfolio:

$$\mu_w = w * \mu_{sp} + (1 - w) * r_f$$

The variance of the profile should be the variance of the linear combination of the weighted portfolio:

$$\sigma^2 = w^2 * \sigma_{sp}^2 + (1 - w)^2 * \sigma_{r_f}^2$$

By definition, the variance of the risk free rate is 0:

$$\sigma^2 = w^2 * \sigma_{sp}^2$$

b

$$U(w) = (w * \mu_{sp} + (1 - w) * r_f) - \frac{\lambda}{2}(w^2 * \sigma_{sp}^2)$$

Maximize by setting the derivative to 0:

$$\frac{dU}{dw} = \mu_{sp} - r_f - \lambda w \sigma_{sp}^2 = 0$$

$$\lambda = \frac{\mu_{sp} - r_f}{w\sigma_{sp}^2}$$

Therefore, if $\lambda = 5$:

$$5 = \frac{\mu_{sp} - r_f}{w\sigma_{sp}^2}$$

$$w = \frac{\mu_{sp} - r_f}{5\sigma_{sp}^2}$$

c

```
mean(diffSp)
var(diffSp)

w <- (mean(diffSp) - (.02/252) )/ (5 * var(diffSp))
print(w)
```

$\mu_{sp} = 2.6509879 \times 10^{-4}$

$\sigma_{sp}^2 = 1.4968278 \times 10^{-4}$

$r_f = 7.9365079 \times 10^{-5}$

d

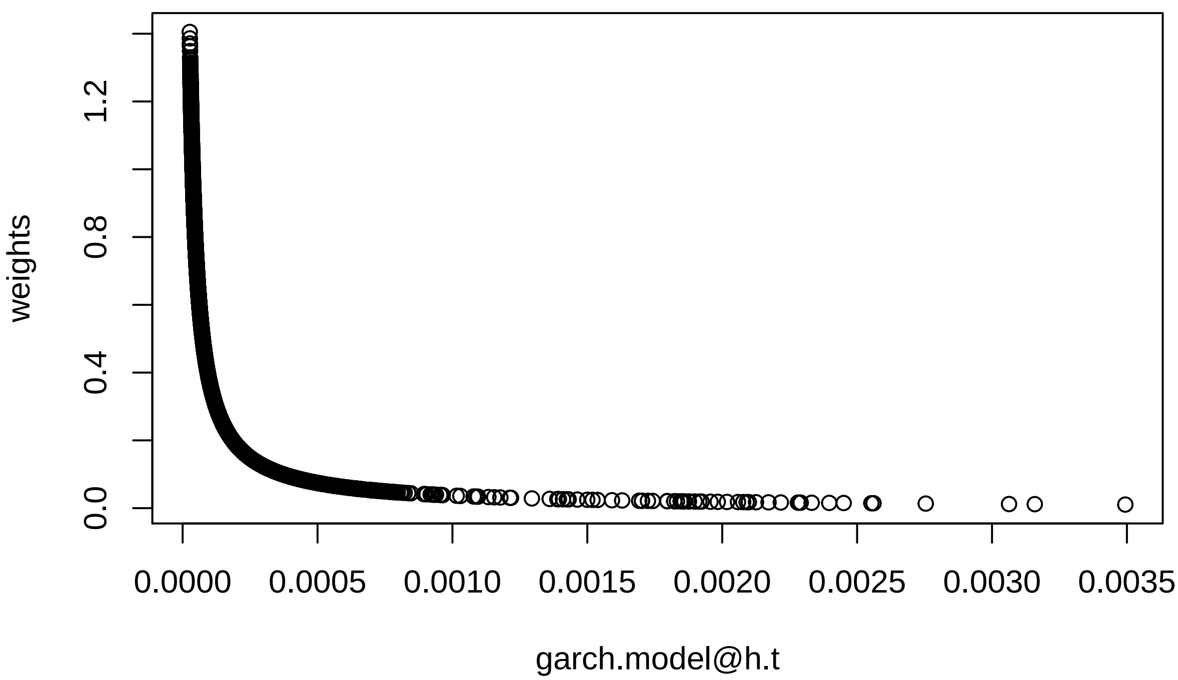
```
w <- (mean(diffSp) - (.02/252) )/ (5 * var(diffSp))
U <- (w * mean(diffSp) - (1-w)*.02/252) - 5/2*(w)
U
```

```
## [1] -0.6204183
```

Problem 11

a

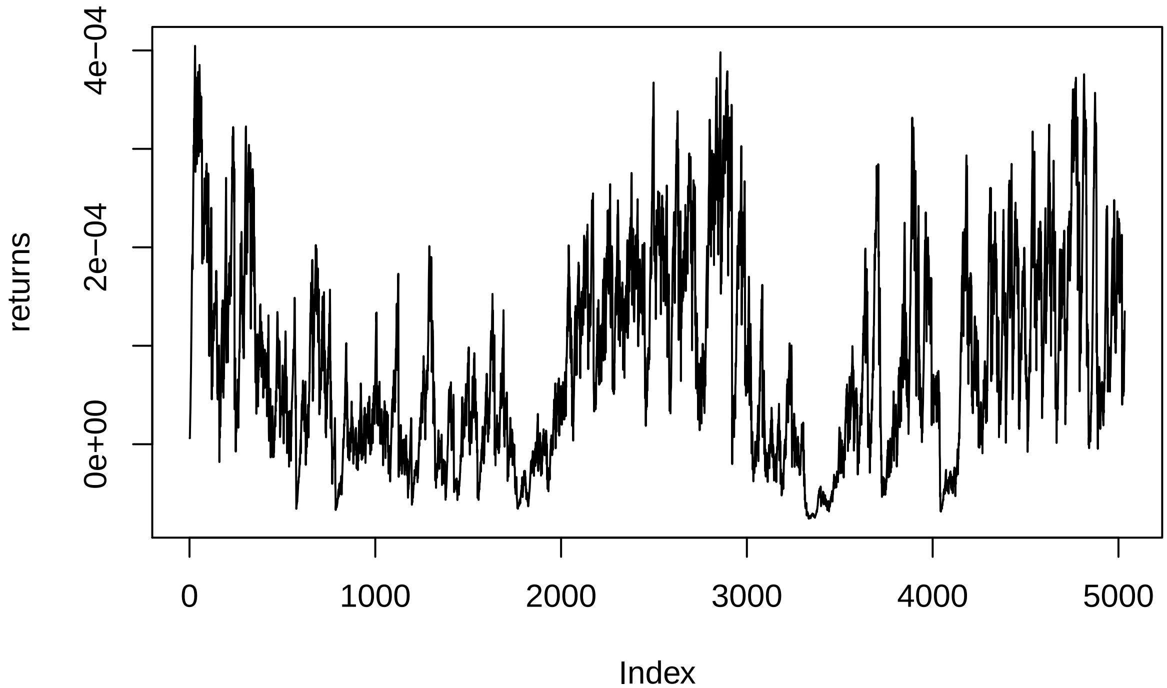
```
w <- (mean(diffSp) - (.02/252) )/ (5 * var(diffSp))

utility <- function(h){
  w*mean(diffSp) + (1-w)*.02/252 - 5/2 * (w^2*h)
}

weight <- function(h){
  (mean(diffSp) - .02/252) / (5 * h)
}

garch.model <- garchFit(~garch(2,2),data=diffSp,trace = F)
weights <-mapply(weight, as.numeric(garch.model@h.t))

plot(garch.model@h.t, weights)
```



Yes, this makes sense because volatility is associated with negative periods of return in the market.

b

```
realized <- function(weight){
  weight * mean(diffSp) - (1 - weight)*(.02/252)
}
returns <- mapply(realized, weights)

plot(returns, type = "l")
```



c

```r
meanRet <- mean(returns)
weightedVar <- 5/2 * sum(returns)

meanRet - weightedVar
```

```
## [1] -1.078284
```

d

```r
us <- returns - 5/2*(returns - meanRet)
sd(us)/sqrt(length(us))
```

```
## [1] 2.153042e-06
```

e

```r
uBar <- mean(returns) - 5/2*sum((returns - meanRet)^2)
print((uBar - U) / (sd(us)/sqrt(length(us))))
```

```
## [1] 288138.1
```

I am clearly rejecting the null, at the moment, (however, due to the lateness) I am going to claim this is a implementation error and with more sleep, I would have written better R.