Homework 5

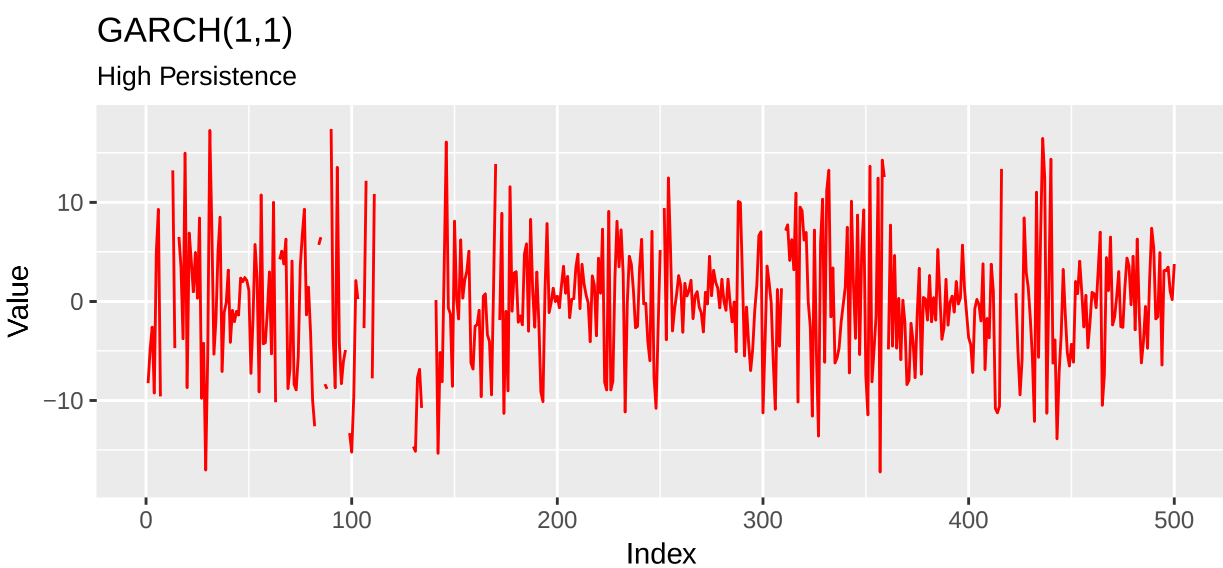
Noah Kawasaki

6/6/2018

Textbook A

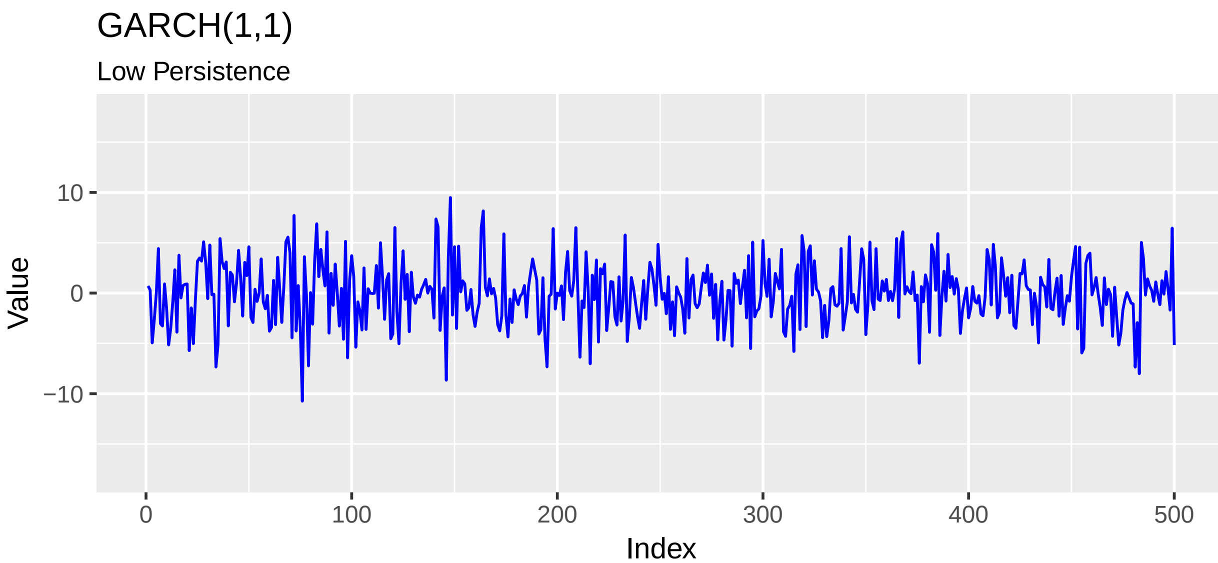
Problem 14.2

```
m1 <- garch.sim(alpha=c(3, 0.35), beta=c(0.6), n=500)  # 0.75
mean1 <- mean(m1)
sd1 <- sd(m1)
sm1 <- (m1-mean1)/sd1
sds1 <- sqrt((m1-mean1)^2)

m2 <- garch.sim(alpha=c(3, 0.1), beta=c(0.6), n=500)  # 0.16
mean2 <- mean(m2)
sd2 <- sd(m2)
sm2 <- (m2-mean2)/sd2
sds2 <- sqrt((m2-mean2)^2)

# Time Series
ggplot(data.frame(m1), aes(x=seq(from=1, to=length(m1)), y=m1)) +
  geom_line(color="red") +
  ylim(-18, 18) +
  ggtitle("GARCH(1,1)", "High Persistence") +
  xlab("Index") +
  ylab("Value")
```

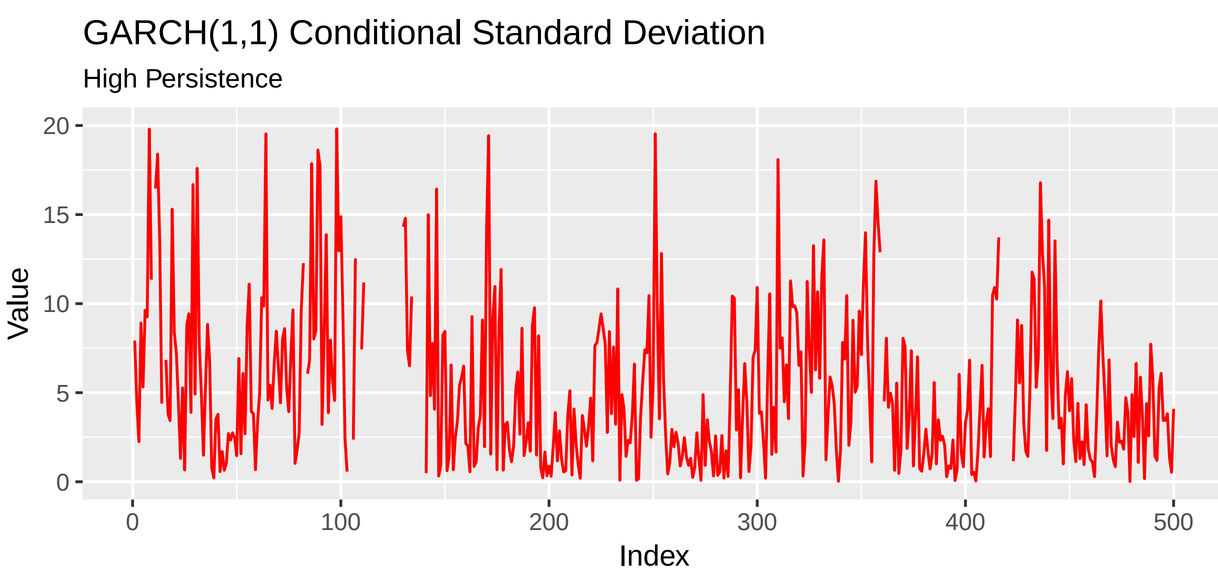



```
ggplot(data.frame(m2), aes(x=seq(from=1, to=length(m2)), y=m2)) +
  geom_line(color="blue") +
  ylim(-18, 18) +
  ggtitle("GARCH(1,1)", "Low Persistence") +
```

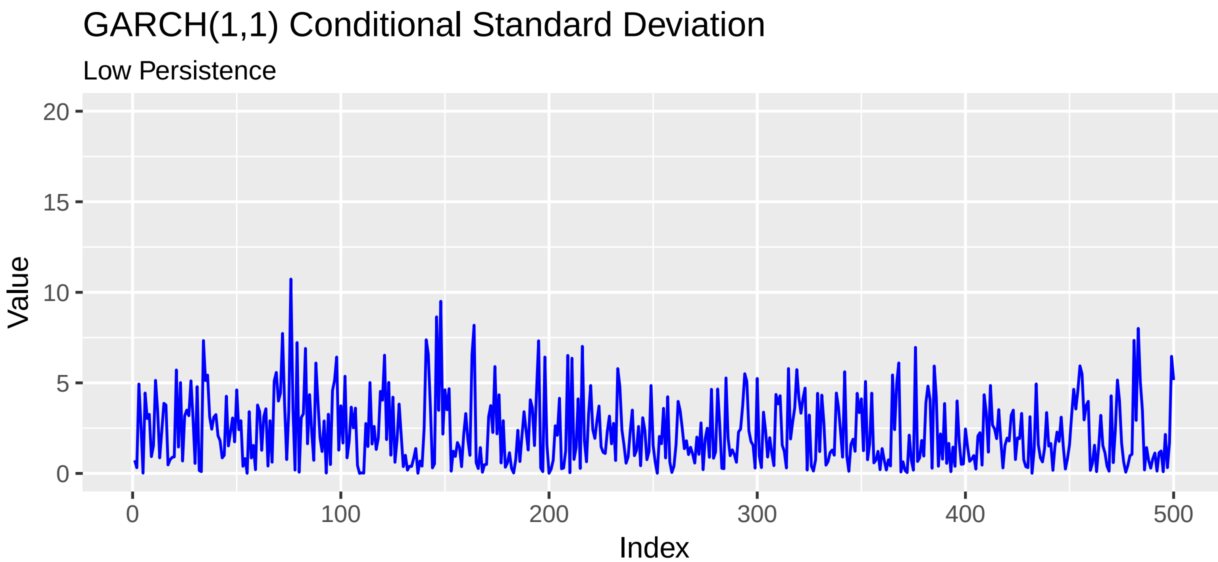
```r
  xlab("Index") +
  ylab("Value")
```




```r
# Conditional Standard Deviations
ggplot(data.frame(m1), aes(x=seq(from=1, to=length(m1)), y=sds1)) +
  geom_line(color="red") +
  ylim(0,20) +
  ggtitle("GARCH(1,1) Conditional Standard Deviation", "High Persistence") +
  xlab("Index") +
  ylab("Value")
```

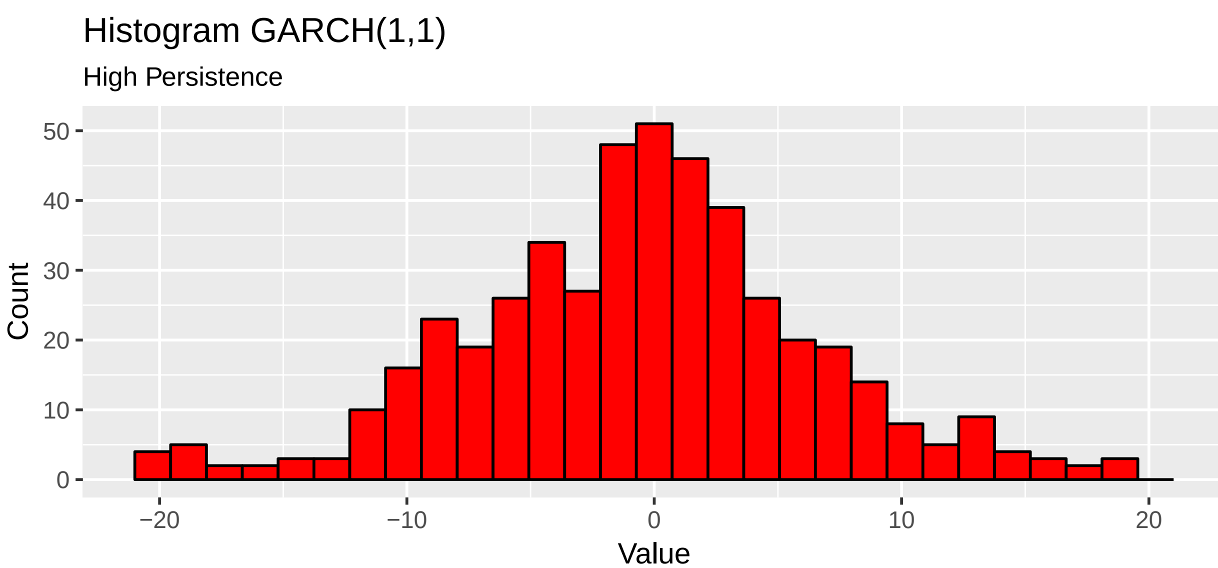



```r
ggplot(data.frame(m1), aes(x=seq(from=1, to=length(m1)), y=sds2)) +
  geom_line(color="blue") +
  ylim(0,20) +
  ggtitle("GARCH(1,1) Conditional Standard Deviation", "Low Persistence") +
  xlab("Index") +
  ylab("Value")
```

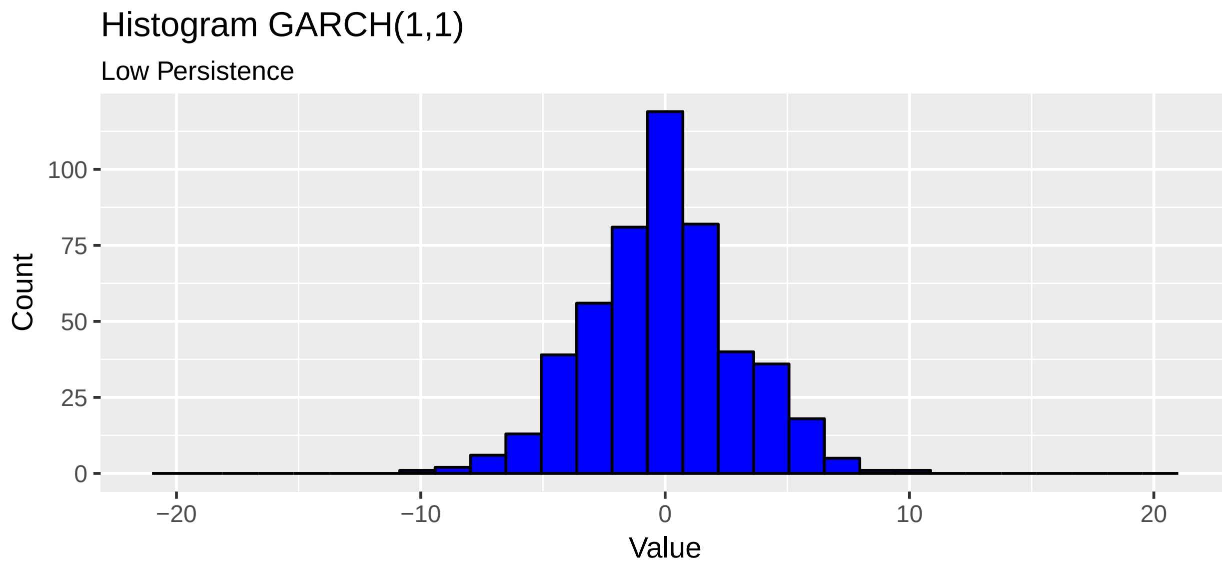


From the simulated time series plots we have a high persistence series with a persistence parameter of 0.875 and a low persistence series with a persistence parameter of 0.25. It is clear that the high persistence series has more volatility clustering and overall volatility, while the lowe persistence series resembles as MA process.

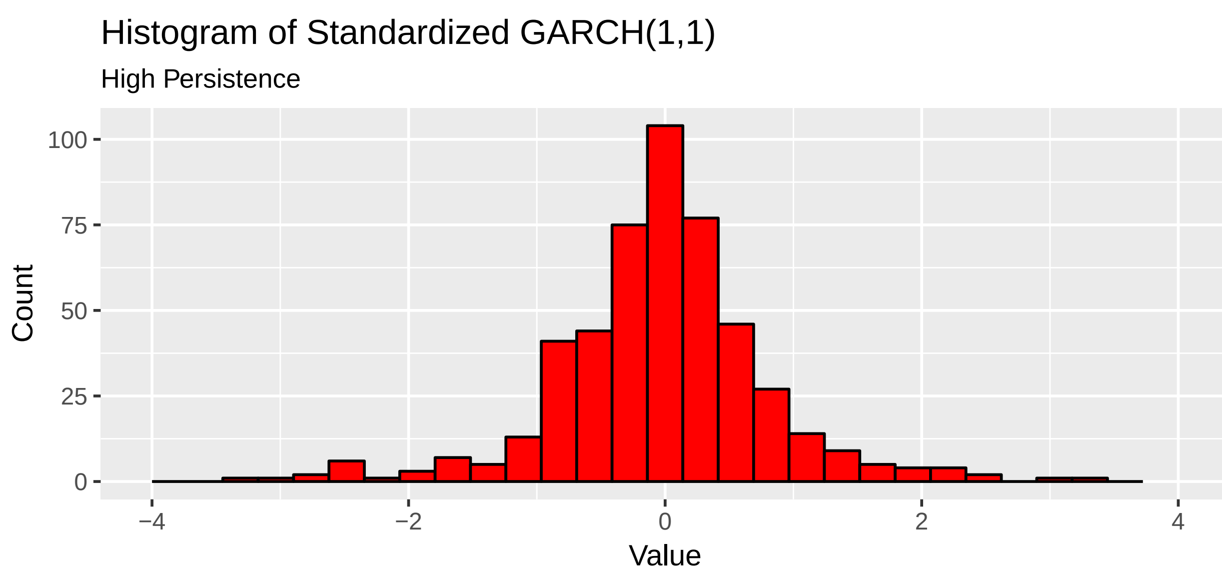
```
# Histogram of original time series
ggplot(data.frame(m1), aes(x=m1)) +
  geom_histogram(fill="red", color="black") +
  xlim(-21, 21) +
  ggtitle("Histogram GARCH(1,1)", "High Persistence") +
  xlab("Value") +
  ylab("Count")
```



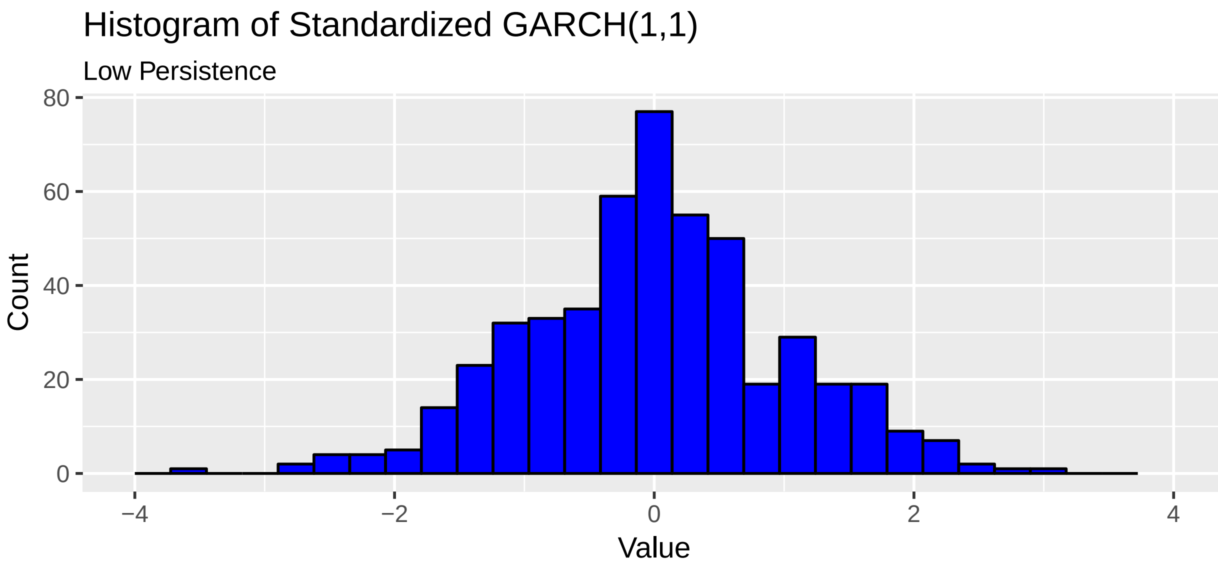

```
ggplot(data.frame(m2), aes(x=m2)) +
  geom_histogram(fill="blue", color="black") +
  xlim(-21, 21) +
  ggtitle("Histogram GARCH(1,1)", "Low Persistence") +
  xlab("Value") +
  ylab("Count")
```




```r
# Histogram of standardized time series
ggplot(data.frame(sm1), aes(x=sm1)) +
  geom_histogram(fill="red", color="black") +
  xlim(-4, 4) +
  ggtitle("Histogram of Standardized GARCH(1,1)", "High Persistence") +
  xlab("Value") +
  ylab("Count")
```

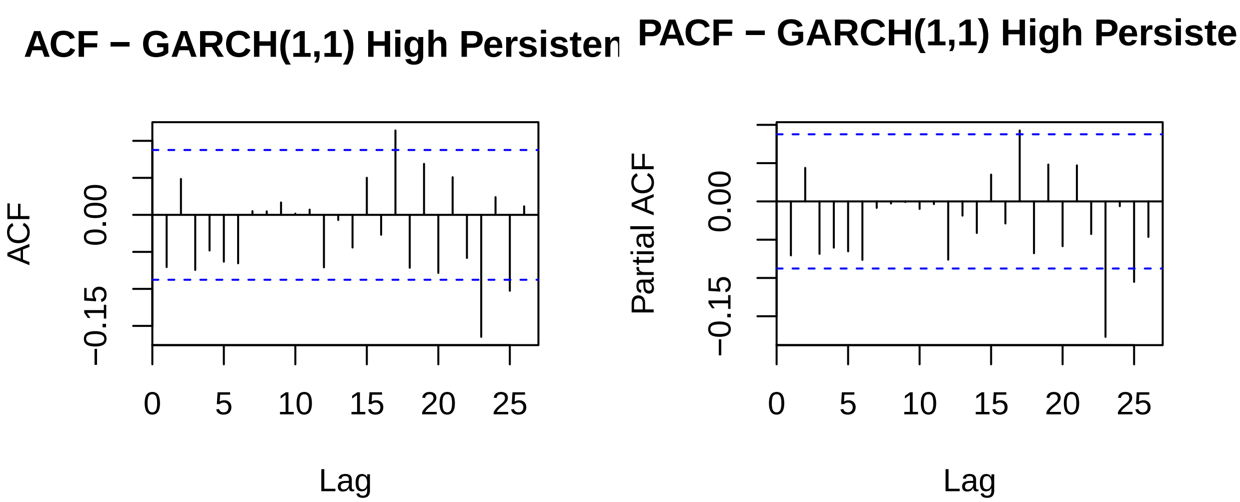



```r
ggplot(data.frame(sm2), aes(x=sm2)) +
  geom_histogram(fill="blue", color="black") +
  xlim(-4, 4) +
  ggtitle("Histogram of Standardized GARCH(1,1)", "Low Persistence") +
  xlab("Value") +
  ylab("Count")
```

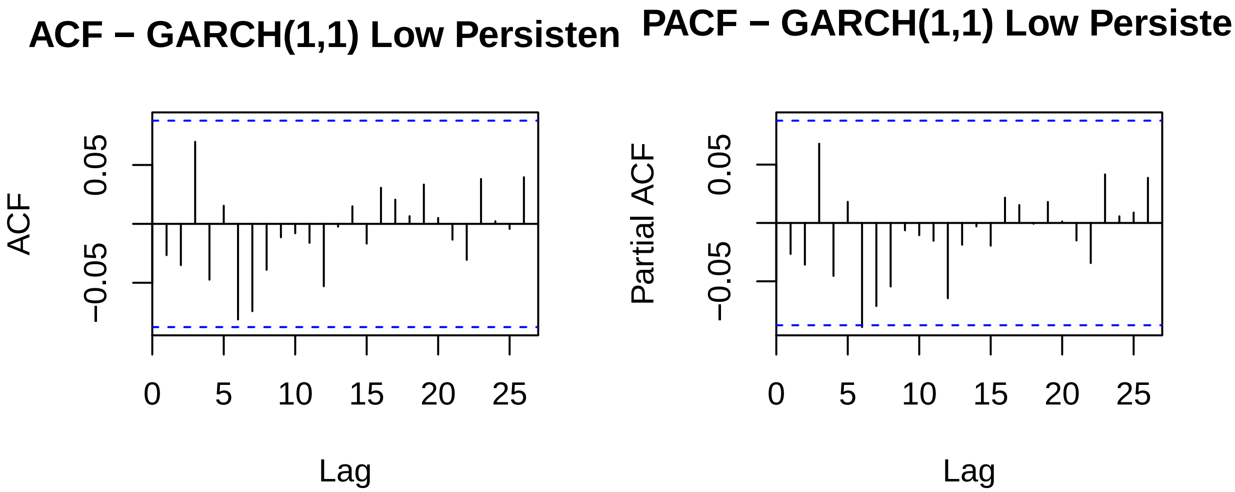


The histogram of the high persistence series shows fatter tails than the low persistence series. This is expected behavior as the persistence causes high and low extreme values to persist over time and thus have higher frequencies. The standardized series histograms exhibit the same patterns but less extremely (because they are standardized).

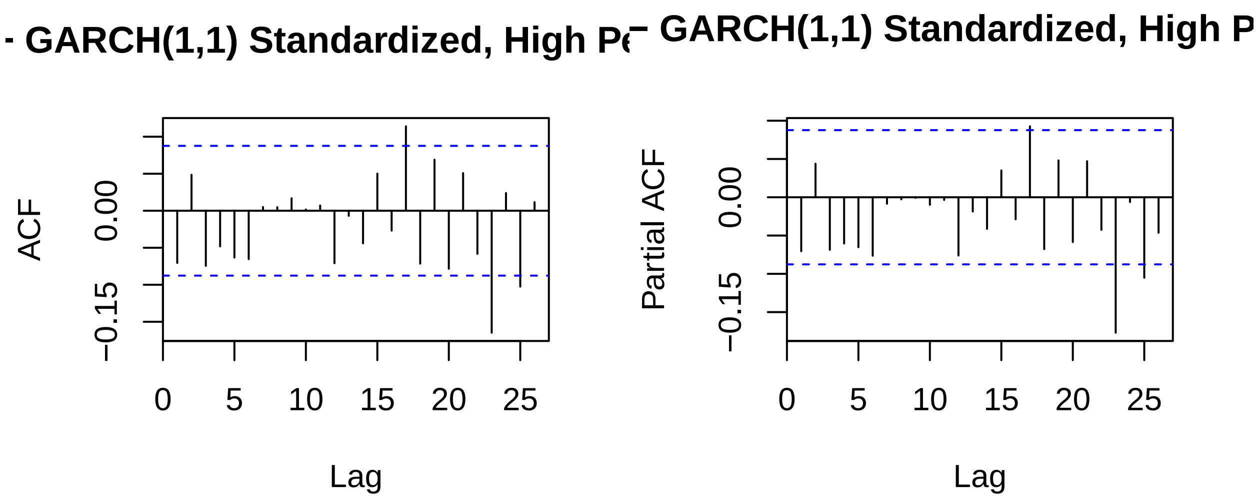
```r
# ACF/PACF original time series
par(mfrow=c(1,2))
acf(m1, main="ACF - GARCH(1,1) High Persistence")
pacf(m1, main="PACF - GARCH(1,1) High Persistence")
```



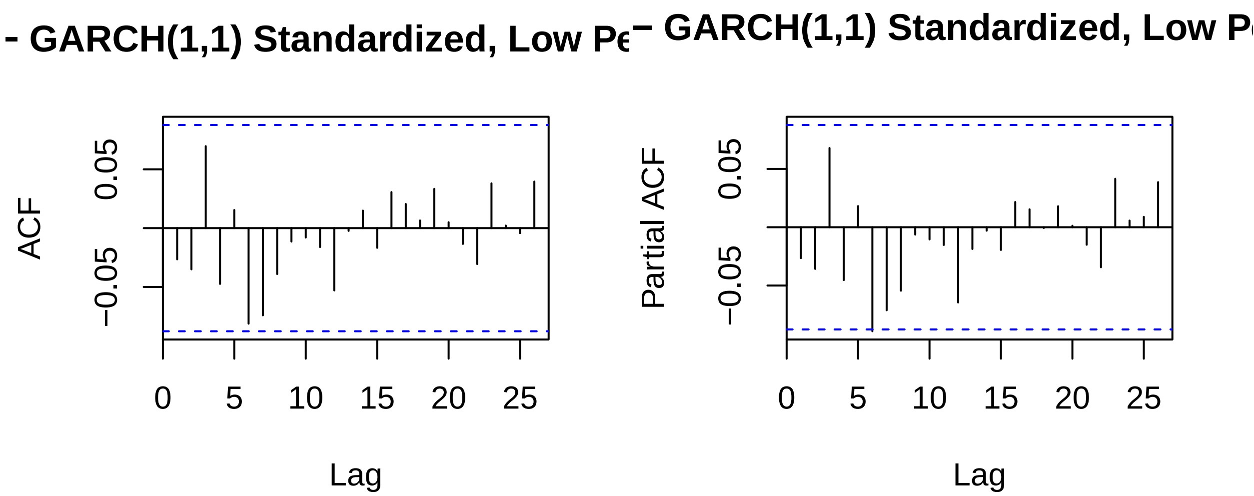

```r
par(mfrow=c(1,2))
acf(m2, main="ACF - GARCH(1,1) Low Persistence")
pacf(m2, main="PACF - GARCH(1,1) Low Persistence")
```




```r
# ACF/PACF standardized series
par(mfrow=c(1,2))
acf(sm1, main="ACF - GARCH(1,1) Standardized, High Persistence")
pacf(sm1, main="PACF - GARCH(1,1) Standardized, High Persistence")
```

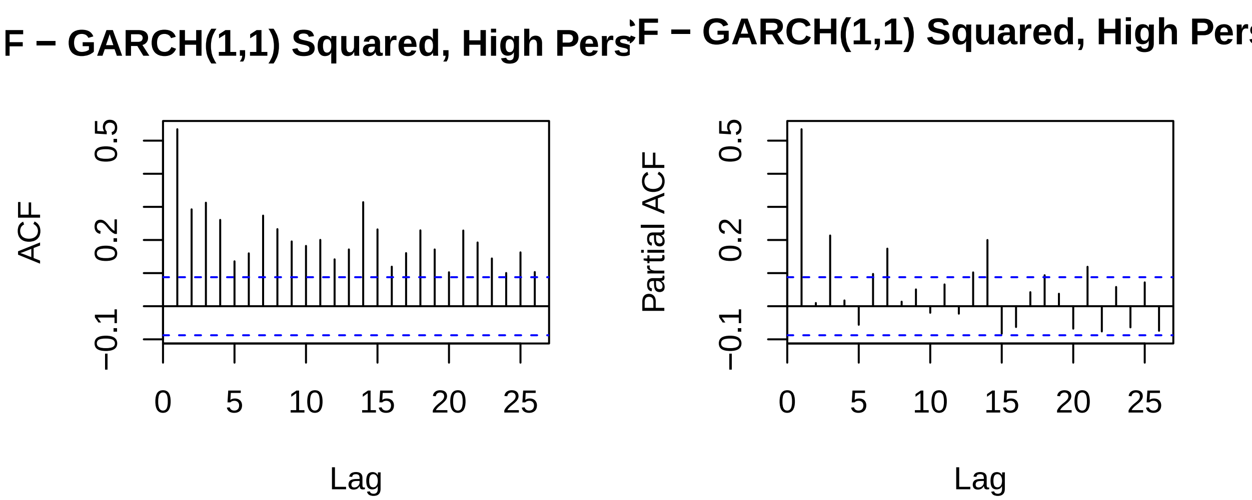



```r
par(mfrow=c(1,2))
acf(sm2, main="ACF - GARCH(1,1) Standardized, Low Persistence")
pacf(sm2, main="PACF - GARCH(1,1) Standardized, Low Persistence")
```

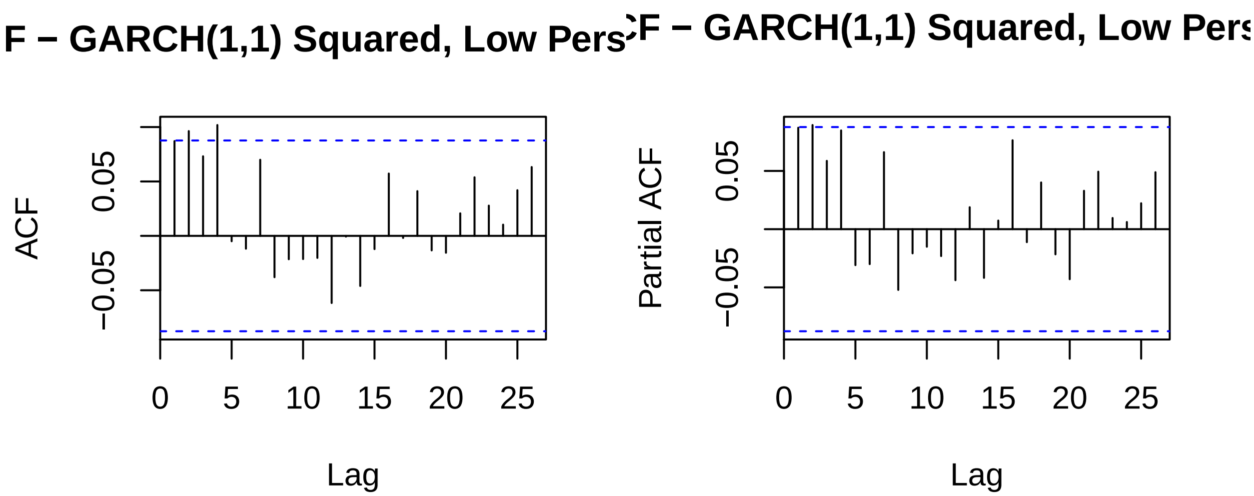


The ACF and PACF's of the original series and standardized original series dont really show any time dependence besides up to a lag of 1 in the high persistence series.

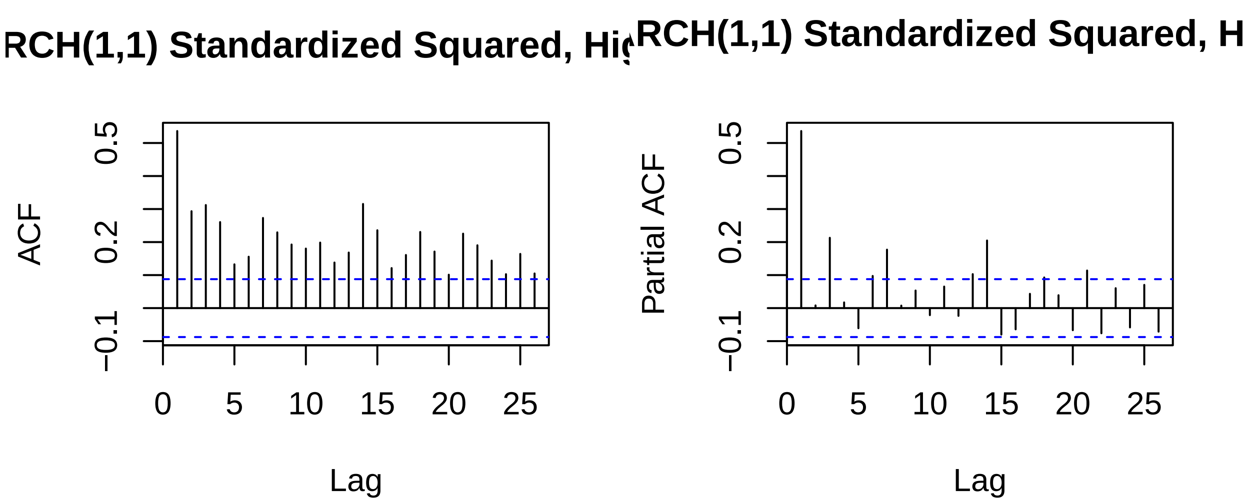
```
# ACF/PACF squared time series
par(mfrow=c(1,2))
acf(m1^2, main="ACF - GARCH(1,1) Squared, High Persistence")
pacf(m1^2, main="PACF - GARCH(1,1) Squared, High Persistence")
```



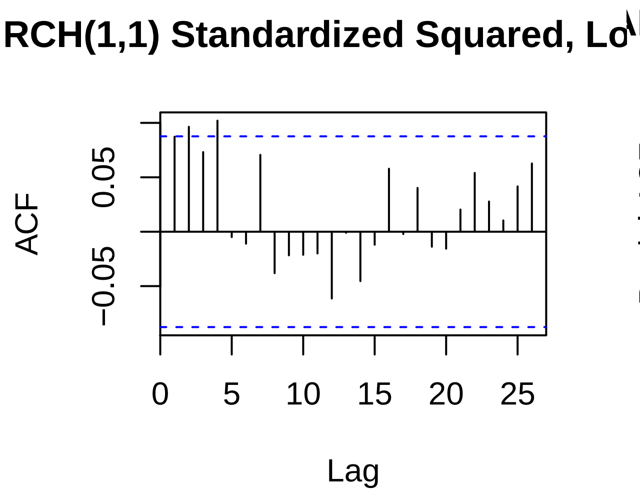

```
par(mfrow=c(1,2))
acf(m2^2, main="ACF - GARCH(1,1) Squared, Low Persistence")
pacf(m2^2, main="PACF - GARCH(1,1) Squared, Low Persistence")
```

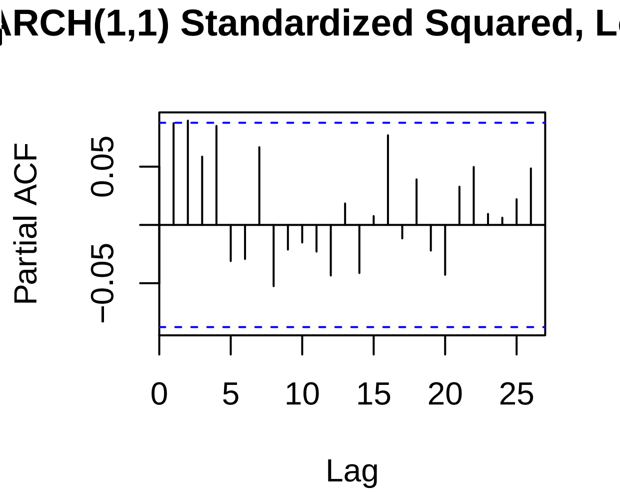



```r
# ACF/PACF squared standardized series
par(mfrow=c(1,2))
acf(sm1^2, main="ACF - GARCH(1,1) Standardized Squared, High Persistence")
pacf(sm1^2, main="PACF - GARCH(1,1) Standardized Squared, High Persistence")
```




```r
par(mfrow=c(1,2))
acf(sm2^2, main="ACF - GARCH(1,1) Standardized Squared, Low Persistence")
pacf(sm2^2, main="PACF - GARCH(1,1) Standardized Squared, Low Persistence")
```



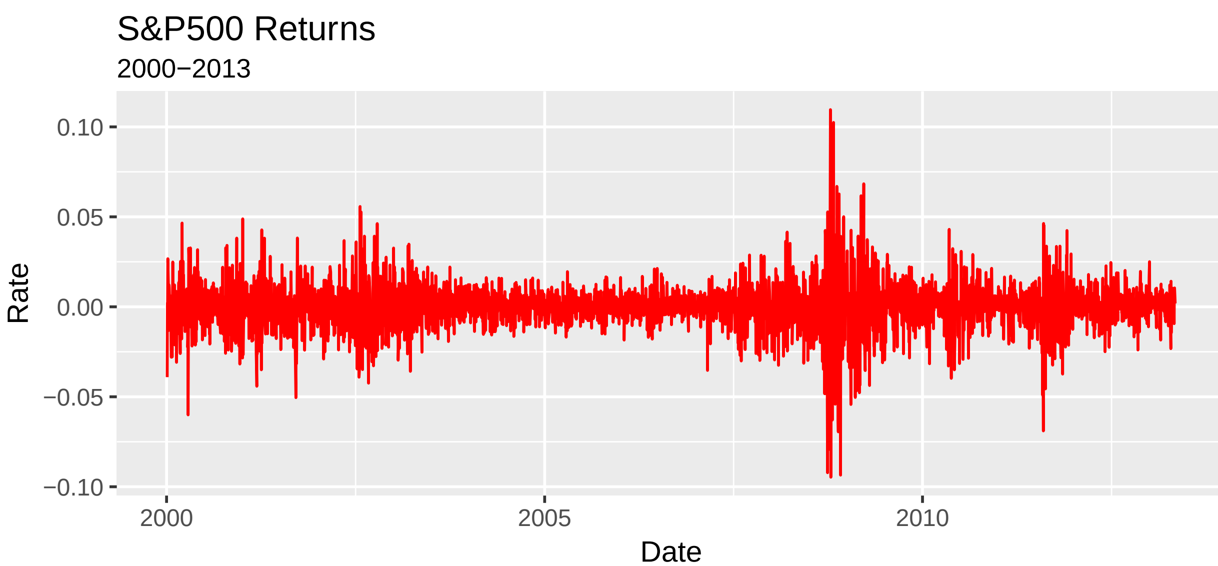


Now, the squared series shows much more time dependence. This is because squaring the series essentially shows us the magnitude of variation from the expected value. Now we see the high persistence series showing some strong time dependence with a slow decay to zero. The low persistence series show only dependence up to one lag.

Problem 14.3

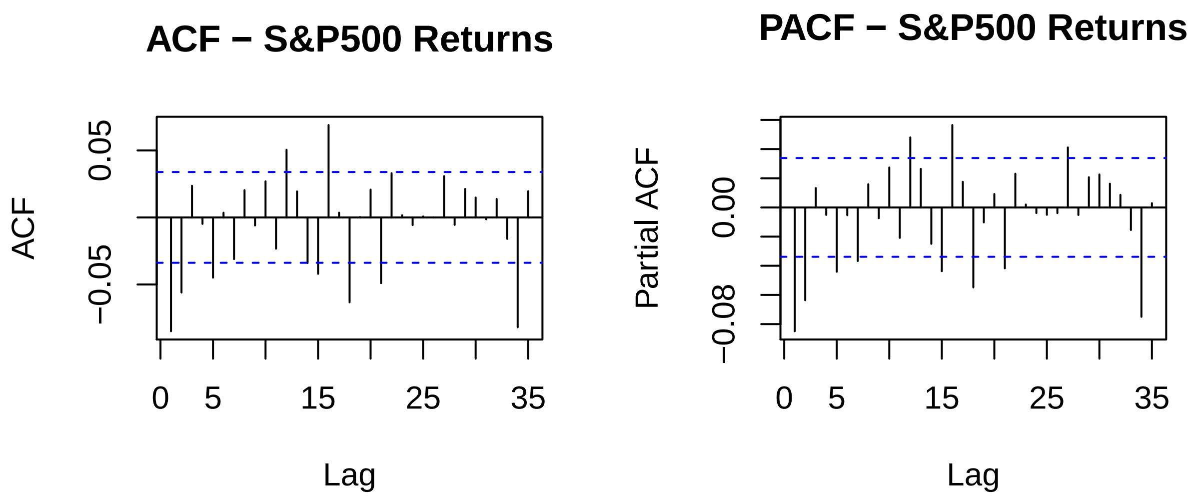
```
sp <- read_xls("/Users/noahkawasaki/Desktop/ECON 144/Textbook_data/Chapter14_exercises_data.xls")
df <- data.frame(sp$Date, sp$`Adj Close`) %>%
  set_names("date", "adjusted") %>%
  mutate(
    return = log(adjusted) - log(lag(adjusted))
  )
df <- na.omit(df)


# Plot time series and return series
ggplot(df, aes(x=date, y=adjusted)) +
  geom_line(color="red") +
  ggtitle("S&P500 Price", "2000-2013") +
  xlab("Date") +
  ylab("Price")
```


```r
ggplot(df, aes(x=date, y=return)) +
  geom_line(color="red", na.rm=TRUE) +
  ggtitle("S&P500 Returns", "2000-2013") +
  xlab("Date") +
  ylab("Rate")
```

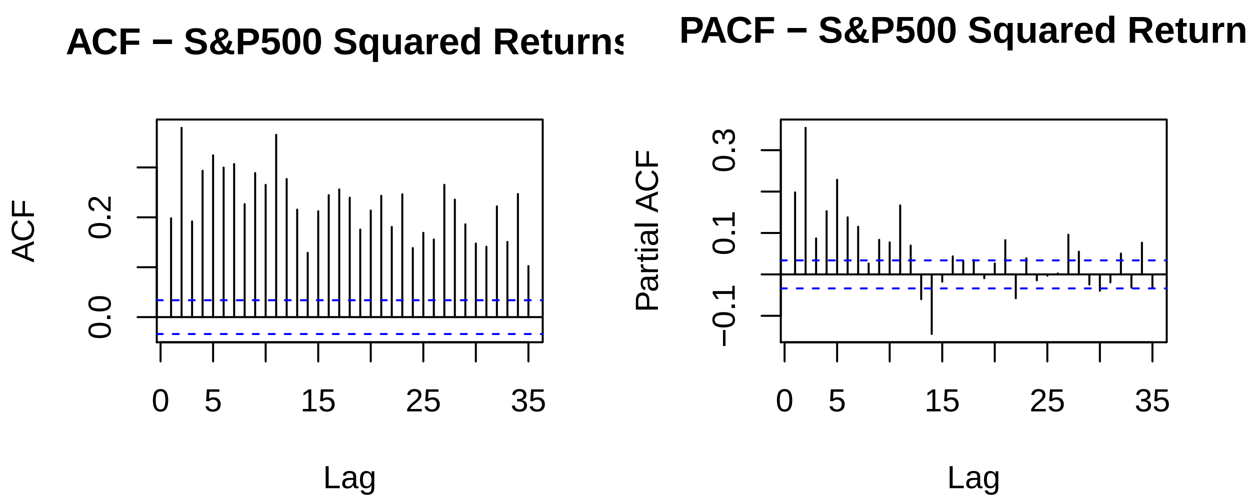


From the time series and returns plot of the S&P 500 it is obvious that this data has periods of high volatility and low volatility. Notably, from 2004 to 2007 is a period of low variation, but from 2007 to 2010 there is some extreme deviation behavior.

```r
# ACF and PACFs
par(mfrow=c(1,2))
acf(df$return, main="ACF - S&P500 Returns")
pacf(df$return, main="PACF - S&P500 Returns")
```




```r
par(mfrow=c(1,2))
acf((df$return)^2, main="ACF - S&P500 Squared Returns")
pacf((df$return)^2, main="PACF - S&P500 Squared Returns")
```



If we look at the ACF and PACF's of the original returns and the squared returns we'll see that there is some variance time dependence, and that ARCH and GARCH models should be applied to tease out these signals.

```r
# Find best ARCH Model
arch = garch(df$return, order=c(0,11), trace=F)
summary(arch)
```

```
##
## Call:
## garch(x = df$return, order = c(0, 11), trace = F)
##
## Model:
## GARCH(0,11)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.66957 -0.56424  0.05792  0.61311  3.23808
##
## Coefficient(s):
```

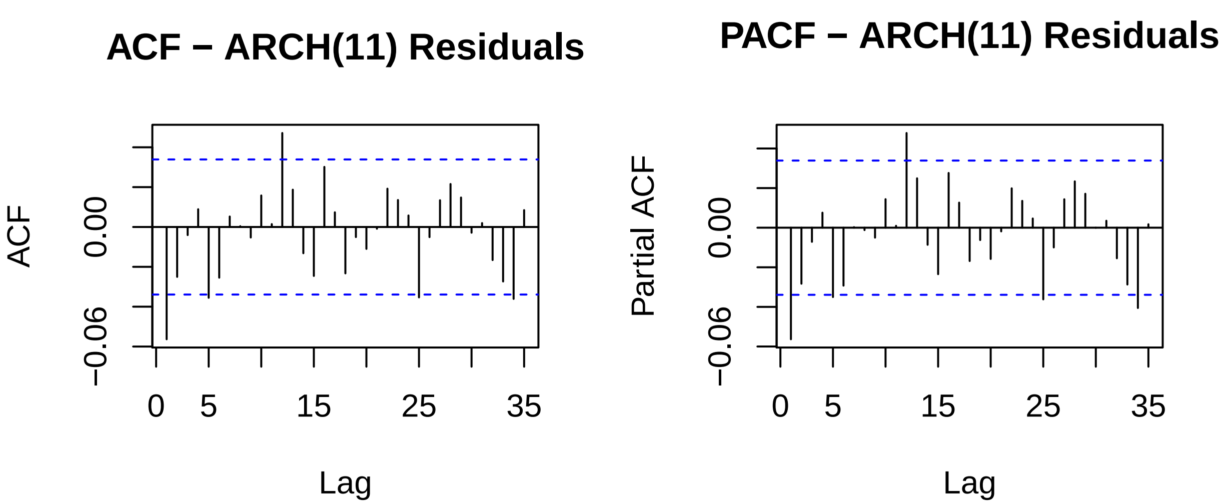
```
##      Estimate  Std. Error  t value Pr(>|t|)
## a0  2.089e-05   1.900e-06   10.994  < 2e-16 ***
## a1  1.117e-02   1.022e-02    1.093 0.274326
## a2  1.369e-01   1.545e-02    8.862  < 2e-16 ***
## a3  8.298e-02   1.836e-02    4.519 6.22e-06 ***
## a4  9.378e-02   1.858e-02    5.046 4.51e-07 ***
## a5  9.358e-02   1.924e-02    4.865 1.15e-06 ***
## a6  5.279e-02   1.877e-02    2.812 0.004929 **
## a7  7.800e-02   1.595e-02    4.889 1.01e-06 ***
## a8  1.172e-01   2.033e-02    5.762 8.29e-09 ***
## a9  6.362e-02   1.731e-02    3.674 0.000239 ***
## a10 8.660e-02   1.889e-02    4.584 4.56e-06 ***
## a11 7.378e-02   1.874e-02    3.936 8.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 203.59, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.03471, df = 1, p-value = 0.8522
```

```r
logLik(arch)
```

```
## 'log Lik.' 10453.14 (df=12)
```

The best ARCH model for this data series is a high order one at ARCH(11). All coefficients but the a1 are statistically significant, and the model achieves a log likelihood of 10453.

```r
par(mfrow=c(1,2))
acf(arch$residuals[12:length(arch$residuals)], main="ACF - ARCH(11) Residuals")
pacf(arch$residuals[12:length(arch$residuals)], main="PACF - ARCH(11) Residuals")
```



Looking at the ACF and PACF of the residuals, it appears that for the most part these signals have been accounted for.

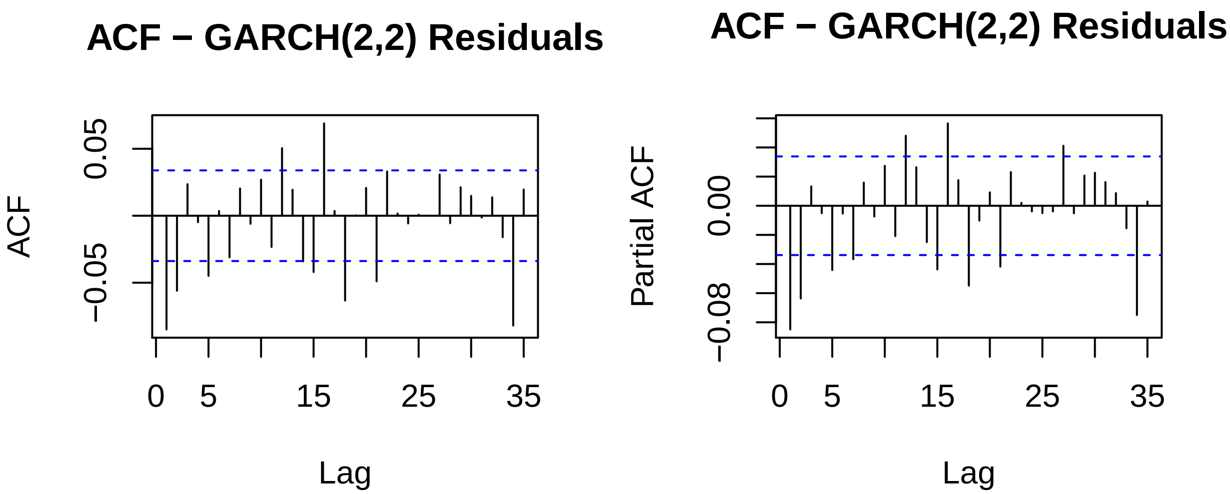
```r
# GARCH Model
garch <- garchFit(~garch(2,2), data=df$return, trace=FALSE)
summary(garch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(2, 2), data = df$return, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(2, 2)
## <environment: 0x7f81293841f0>
##  [data = df$return]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1      alpha2       beta1       beta2
## 3.1522e-04  2.6470e-06  1.2701e-02  1.2445e-01  5.5700e-01  2.8875e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     3.152e-04   1.530e-04    2.060  0.03943 *
## omega  2.647e-06   5.786e-07    4.575 4.76e-06 ***
## alpha1 1.270e-02   1.298e-02    0.978  0.32793
## alpha2 1.245e-01   2.060e-02    6.040 1.54e-09 ***
## beta1  5.570e-01   1.718e-01    3.241  0.00119 **
## beta2  2.887e-01   1.575e-01    1.834  0.06669 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  10489.13    normalized:  3.127351
##
## Description:
##  Thu Jun  7 14:40:31 2018 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  238.1684  0
##  Shapiro-Wilk Test  R    W      0.9892104 2.722248e-15
##  Ljung-Box Test     R    Q(10)  22.1942   0.01414527
##  Ljung-Box Test     R    Q(15)  32.07282  0.006292776
##  Ljung-Box Test     R    Q(20)  38.11685  0.00856893
##  Ljung-Box Test     R^2  Q(10)  6.155105  0.8020679
##  Ljung-Box Test     R^2  Q(15)  9.255884  0.8637624
```

```
##  Ljung-Box Test      R^2  Q(20)  12.06721  0.9137434
##  LM Arch Test        R    TR^2   6.430372  0.8928536
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -6.251124 -6.240180 -6.251130 -6.247210
```

While the ARCH(11) did a good job at modeling the S&P returns, it contains 12 parameters to estimate. So we can also fit a GARCH model to achieve similar (better?) results with less computing. The chosen model is GARCH(2,2). Its coefficients are statistically significant besides the alpha1, and gets a higher log likelihood of 10487.

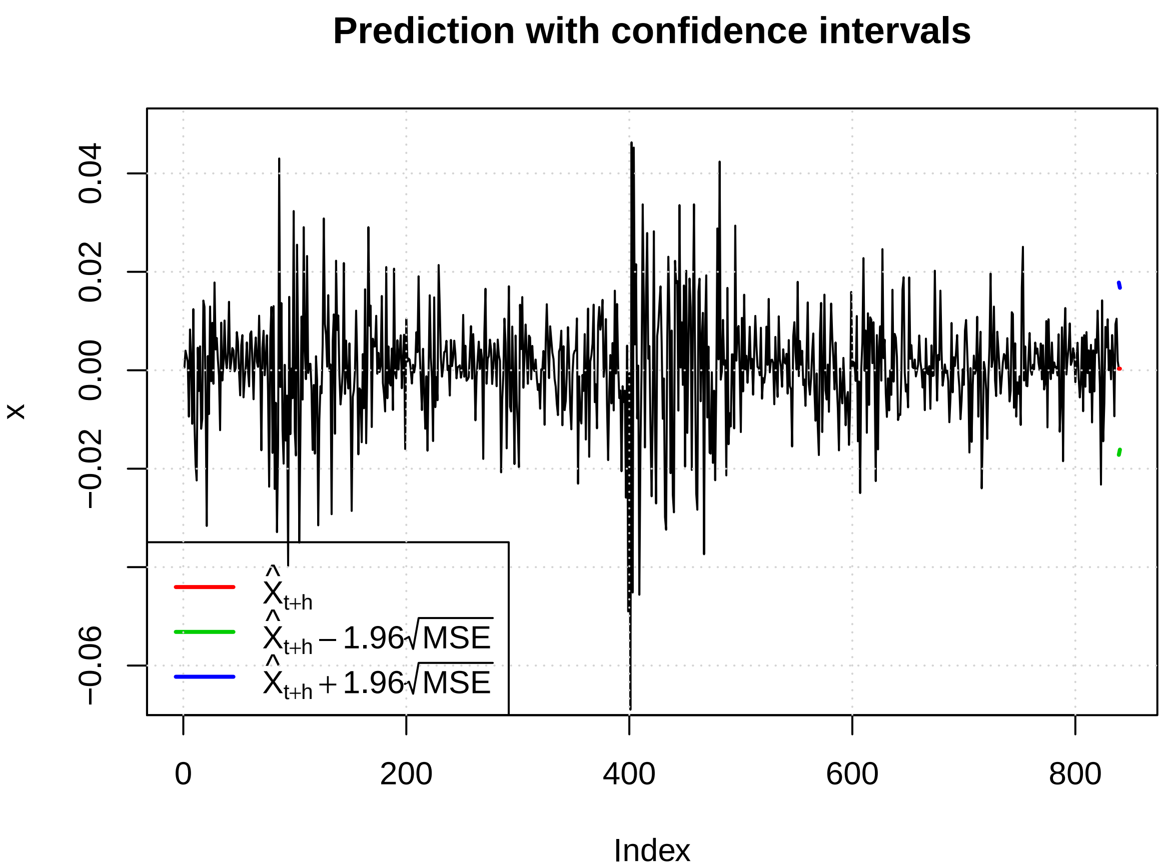
```r
par(mfrow=c(1,2))
acf(garch@residuals, main="ACF - GARCH(2,2) Residuals")
pacf(garch@residuals, main="ACF - GARCH(2,2) Residuals")
```



The ACF and PACF of the GARCH(2,2) residuals also suggest that the model effectively captured the volatile behavior.

Problem 14.4

```r
# One and two step ahead forecasts
predicts <- predict(garch, n.ahead=2, plot=TRUE)
```




```
predicts
```

```
##   meanForecast   meanError standardDeviation lowerInterval upperInterval
## 1 0.0003152196 0.008928532       0.008928532   -0.01718438    0.01781482
## 2 0.0003152196 0.008387256       0.008387256   -0.01612350    0.01675394
```

Textbook B

Problem 14.1

```
## a
getSymbols("^NYA", from="1988-01-01", to="2001-12-31")
```

```
## [1] "NYA"
```

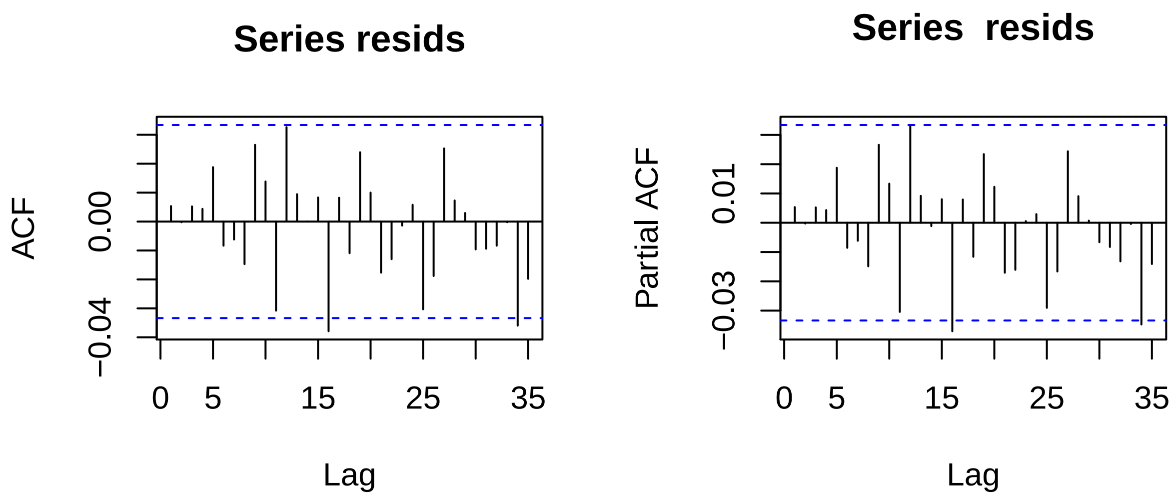
```
nyse <- data.frame(log(NYA$NYA.Adjusted) - log(lag(NYA$NYA.Adjusted))[-1])
nyse <- tibble::rownames_to_column(nyse) %>%
  set_names("date", "return")

train <- nyse[1:3461, ]
test <- nyse[3462:nrow(nyse), ]

ts <- ts(train$return)
ts_test <- ts(test$return)

# Plot
```

```r
ggplot(train, aes(x=seq(from=1, to=length(return)), y=return, group=1)) +
  geom_line(color="green") +
  ggtitle("NYSE Daily Returns", "1988-2001") +
  xlab("Index") +
  ylab("Rate")
```

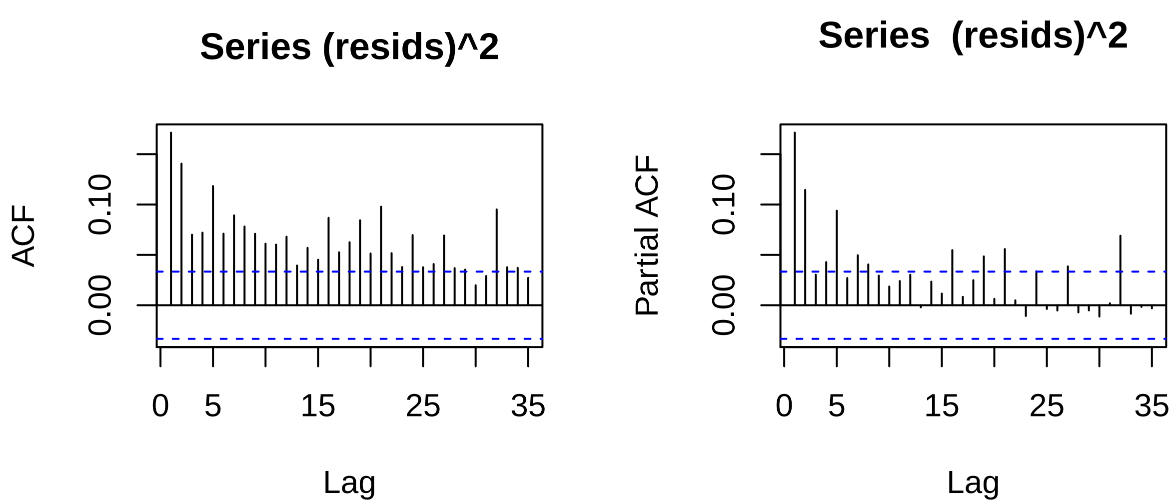

```r
# AR
m = auto.arima(ts)
summary(m)
```

```
## Series: ts
## ARIMA(5,0,3) with non-zero mean
##
## Coefficients:
##          ar1     ar2      ar3     ar4      ar5      ma1      ma2     ma3
##       0.8503  0.3857  -0.6647  0.0736  -0.0526  -0.8023  -0.4509  0.6292
## s.e.  0.1601  0.2409   0.1552  0.0244   0.0232   0.1599   0.2360  0.1454
##         mean
##       4e-04
## s.e.  1e-04
##
## sigma^2 estimated as 7.241e-05:  log likelihood=11590.62
## AIC=-23161.25   AICc=-23161.18   BIC=-23099.75
##
## Training set error measures:
##                         ME        RMSE        MAE MPE MAPE      MASE
## Training set -1.62577e-06 0.008498468 0.00598073 NaN  Inf 0.7023232
##                       ACF1
## Training set -0.0001363117
```

```r
resids <- m$residuals[13:length(m$residuals)]

par(mfrow=c(1,2))
acf(resids)
pacf(resids)
```




```r
par(mfrow=c(1,2))
acf((resids)^2)
pacf((resids)^2)
```

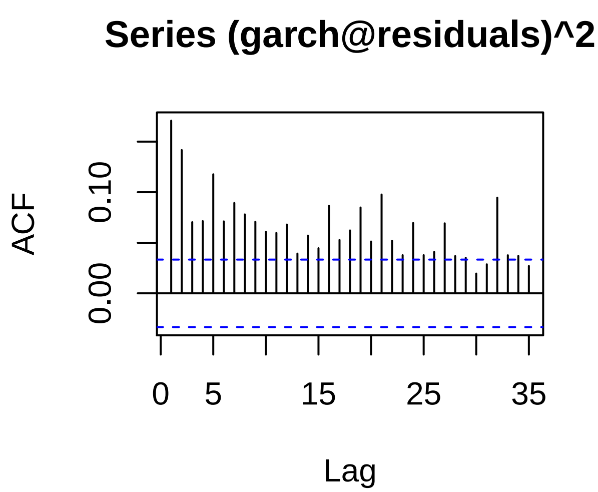


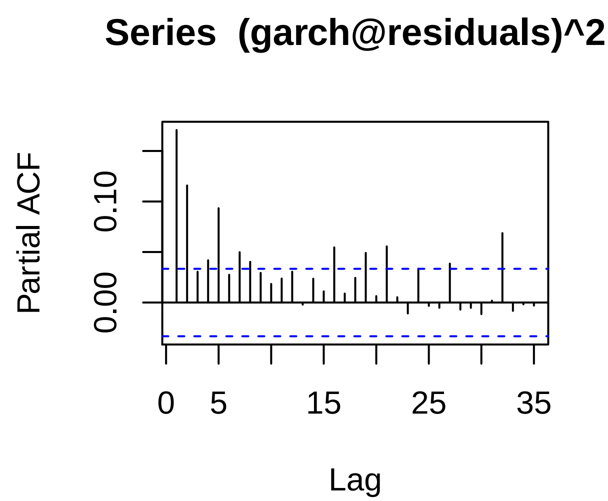

```r
# GARCH
garch <- garchFit(~garch(1,2), data=resids, trace=FALSE)
summary(garch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 2), data = resids, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 2)
## <environment: 0x7f812d139d80>
##  [data = resids]
##
## Conditional Distribution:
##  norm
##
```

```
## Coefficient(s):
##         mu       omega      alpha1       beta1       beta2
## 1.0776e-04  6.8337e-07  5.1809e-02  9.1626e-01  2.4135e-02
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     1.078e-04   1.213e-04    0.889  0.37420
## omega  6.834e-07   2.300e-07    2.971  0.00297 **
## alpha1 5.181e-02   1.020e-02    5.082 3.74e-07 ***
## beta1  9.163e-01   1.545e-01    5.930 3.03e-09 ***
## beta2  2.414e-02   1.489e-01    0.162  0.87123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  11860.31    normalized:  3.438768
##
## Description:
##  Thu Jun  7 14:40:37 2018 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  5229.4    0
##  Shapiro-Wilk Test  R    W      0.9595893 0
##  Ljung-Box Test     R    Q(10)  10.20268  0.4228949
##  Ljung-Box Test     R    Q(15)  15.18456  0.4382065
##  Ljung-Box Test     R    Q(20)  18.0473   0.584292
##  Ljung-Box Test     R^2  Q(10)  3.603135  0.9634799
##  Ljung-Box Test     R^2  Q(15)  5.478519  0.9872411
##  Ljung-Box Test     R^2  Q(20)  6.373636  0.9982896
##  LM Arch Test       R    TR^2   4.75903   0.9655469
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -6.874637 -6.865728 -6.874642 -6.871455
```

```r
par(mfrow=c(1,2))
acf((garch@residuals)^2)
pacf((garch@residuals)^2)
```

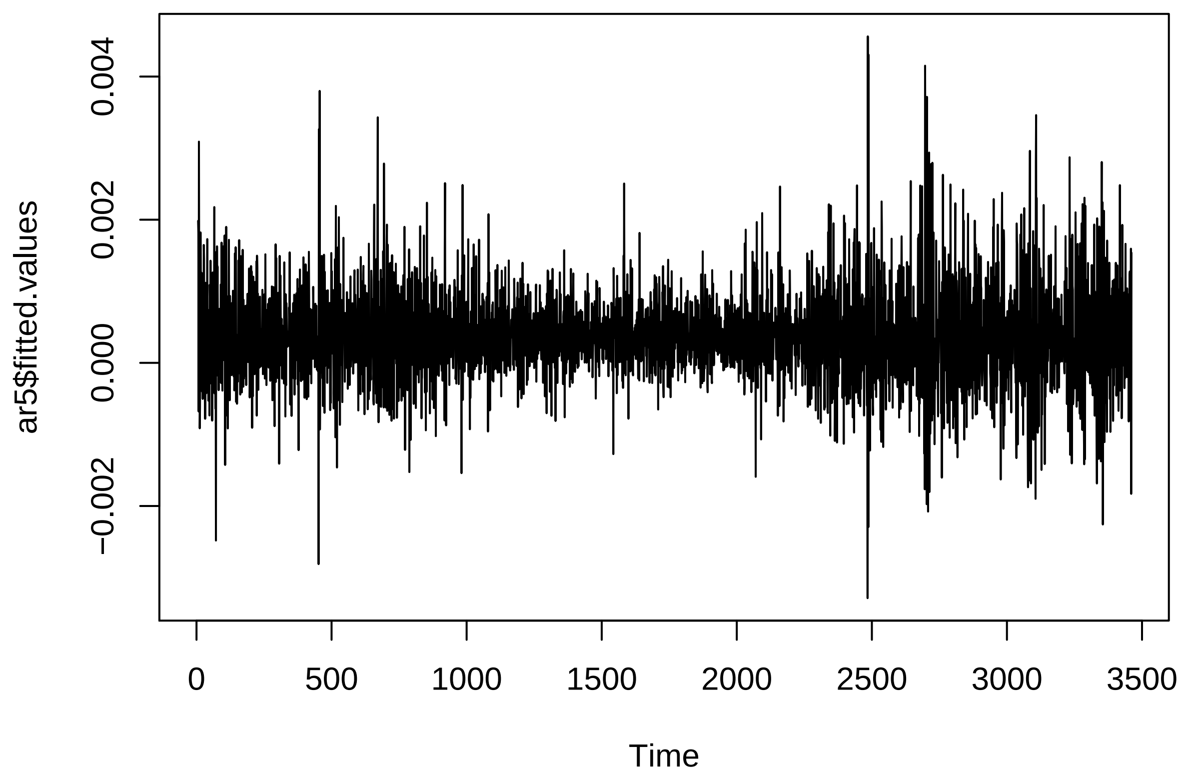




Problem 14.4

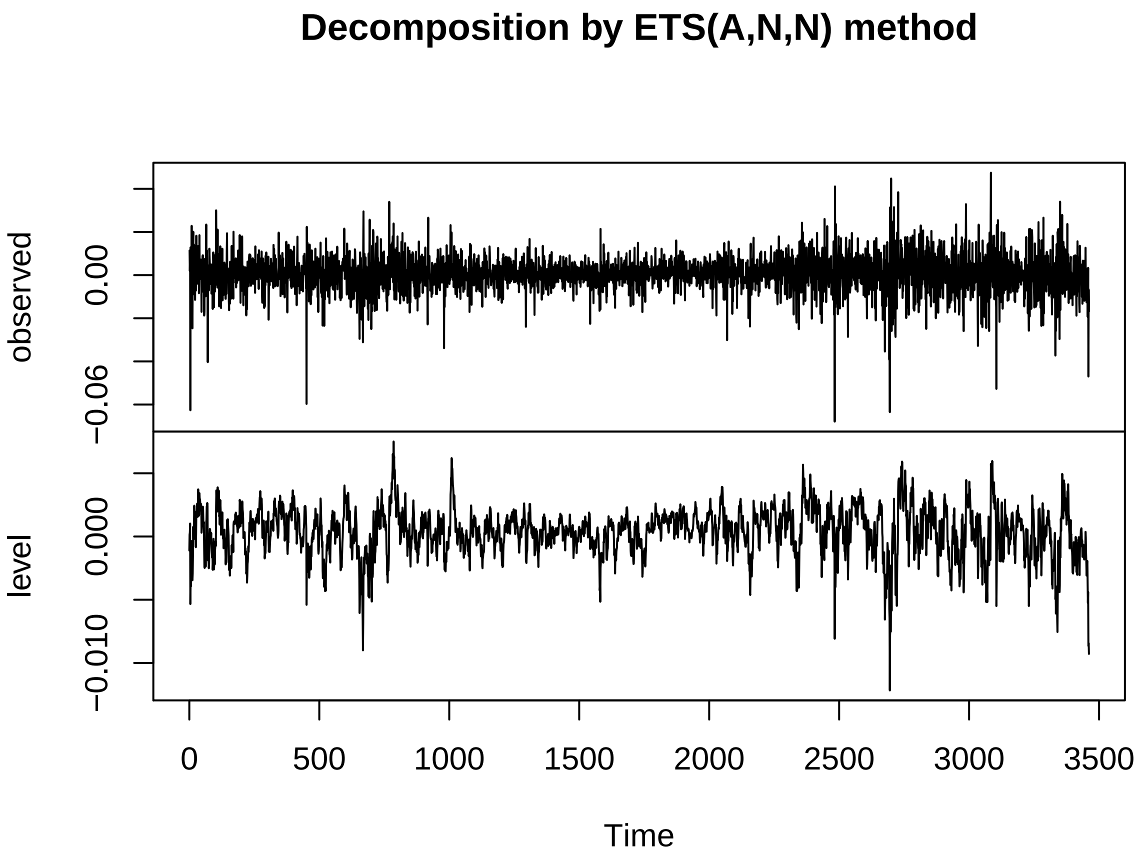
a)

```r
ar5 <- arma(ts, order=c(5,0))
plot(ar5$fitted.values)
```



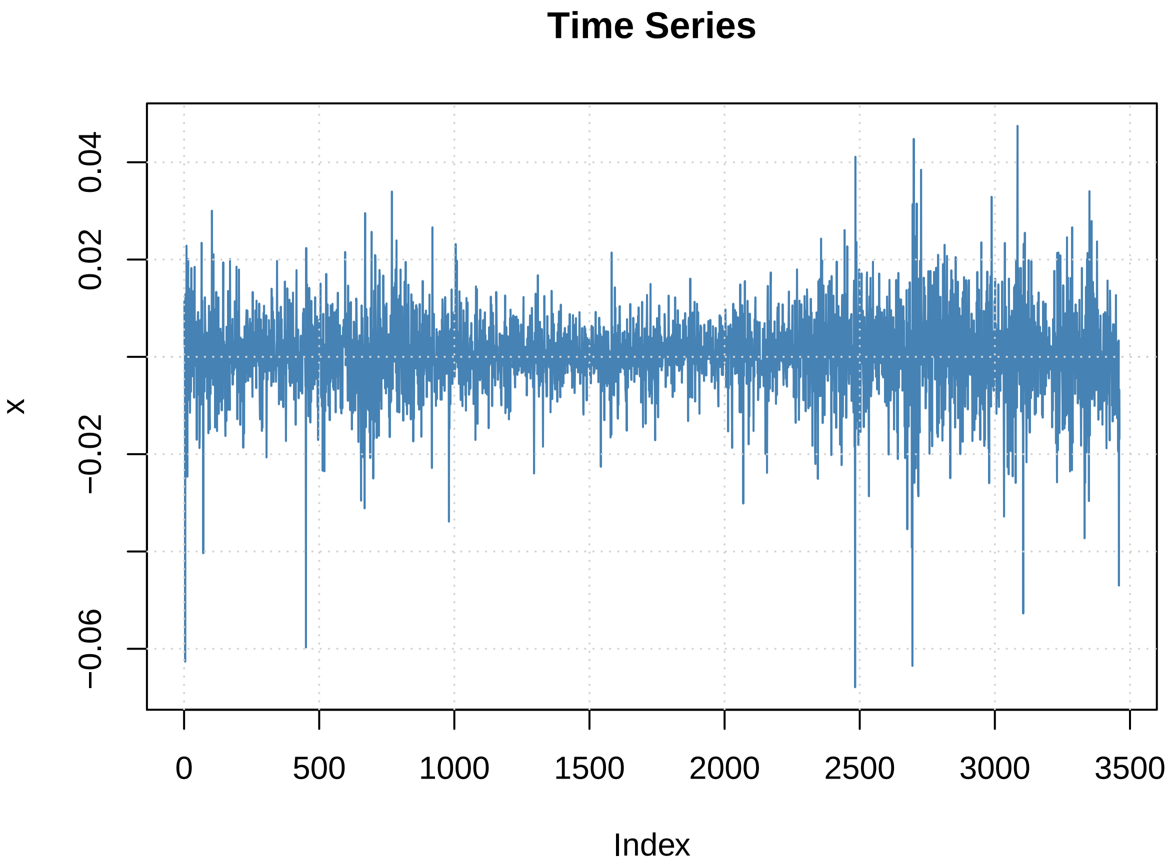
b)

```r
smooth <- ets(ts, model="ANN", alpha=0.10)
plot(smooth)
```



c)

```r
garch11 <- garchFit(~garch(1,2), data=ts, trace=FALSE)
plot(garch11, which=1)
```



d)

For the most part, visually these models all look very similar. However the ARCH and GARCH model's are based on actual time dependence as opposed to the mathematical strategies used in exponential smoothing. This means that the GARCH's smoothing parameter is chosen through likelihood optimization, whereas we just chose an arbitrary value for the ets model.

Problem 14.5

a)

```
spec <- garchSpec(model=list(mu=76, alpha=0.6, beta=0, omega=3))
sim <- garchSim(spec, n=100)

fit <- garchFit(~garch(1, 1), sim, trace=FALSE)
predict(fit, n.ahead=10)
```

```
##    meanForecast meanError standardDeviation
## 1      76.19167  1.854680          1.854680
## 2      76.19167  2.051320          2.051320
## 3      76.19167  2.163883          2.163883
## 4      76.19167  2.230583          2.230583
## 5      76.19167  2.270807          2.270807
## 6      76.19167  2.295304          2.295304
```

```
## 7      76.19167  2.310307          2.310307
## 8      76.19167  2.319526          2.319526
## 9      76.19167  2.325203          2.325203
## 10     76.19167  2.328703          2.328703
```

b)

c)

Answer is not necessarily. Because a GARCH model is based on the time dependence of volatility and is short term, in order to predict a next spell of bad weather that bad weather would have already needed to start.