HW7 - CFRM502 - Winter 2022

David Long (9426749)

2/28/2022

Setup

```
library('forecast')
library('fracdiff')
library('fGarch')
library('MASS')
load("Homework 7 Data.Rdata")
attach(futures)
```

Question 1

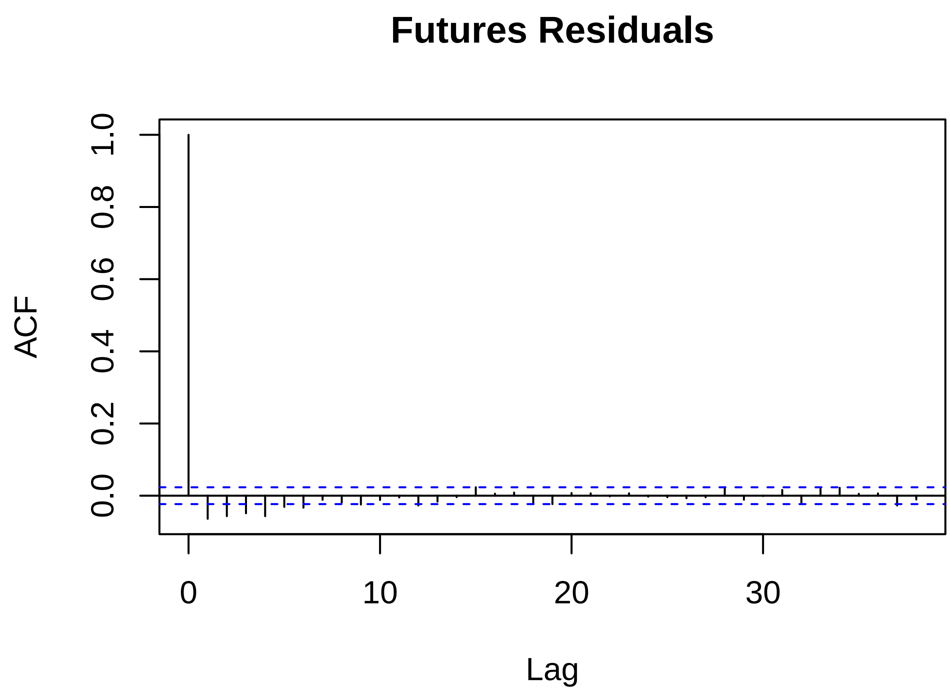
(a) Residual Analysis of Linear Fit

```
futures.lm <- lm(diff(lnfuture) ~ diff(lnspot))
summary(futures.lm)
```

```
##
## Call:
## lm(formula = diff(lnfuture) ~ diff(lnspot))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0038484 -0.0001568 -0.0000014  0.0001612  0.0026256
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.354e-06  3.509e-06   0.386      0.7
## diff(lnspot) 6.212e-01  1.754e-02  35.420   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0002948 on 7058 degrees of freedom
## Multiple R-squared:  0.1509, Adjusted R-squared:  0.1508
## F-statistic:  1255 on 1 and 7058 DF,  p-value: < 2.2e-16
```

The fitted model is $Y_i = 0.62122X_i + \varepsilon_i$. The intercept is not significant.

```
futures.res <- resid(futures.lm)
acf(futures.res, main = 'Futures Residuals')
```




```
Box.test(futures.res, lag = 20, type = 'Ljung-Box')
```

```
##
##  Box-Ljung test
##
## data:  futures.res
## X-squared = 135.02, df = 20, p-value < 2.2e-16
```

The residuals are correlated. This is evidenced by the first six lags in the ACF plot, which exceed the confidence boundaries, and the Ljung-Box test, which has a $P-$value of 2.2e-16, which leads to a rejection of the null hypothesis that all autocorrelations are zero.

(b) ARIMAX Fit

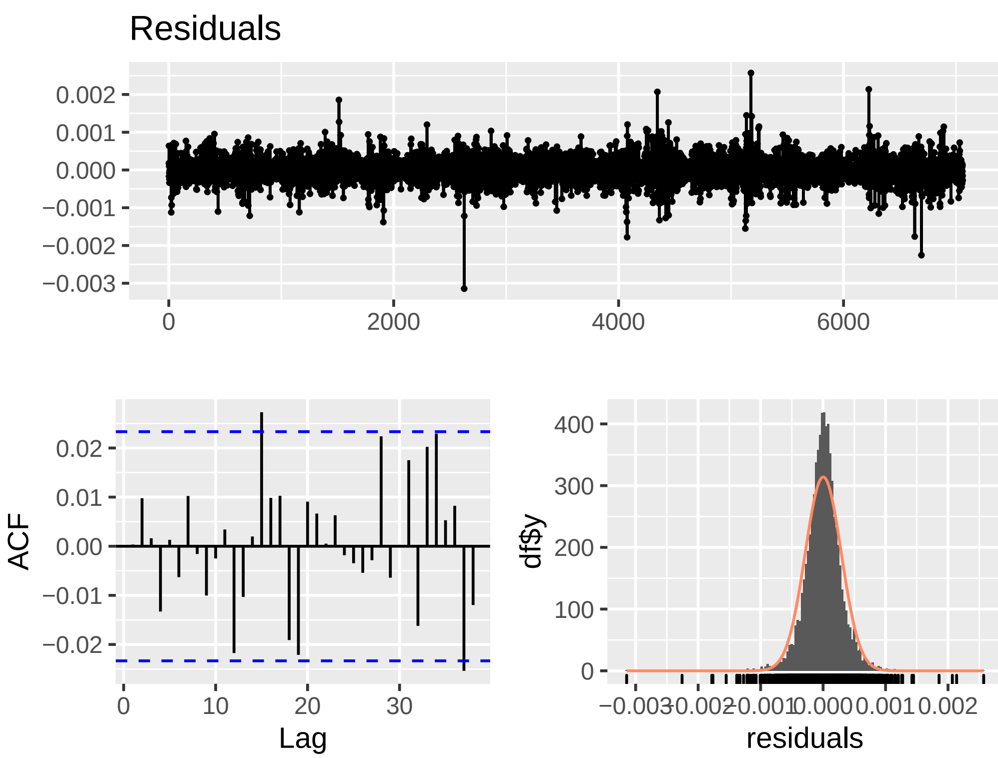
```
futures.auto.arimax <- auto.arima(diff(lnfuture), xreg = diff(lnspot))
futures.auto.arimax
```

```
## Series: diff(lnfuture)
## Regression with ARIMA(1,0,2) errors
##
## Coefficients:
##          ar1      ma1      ma2    xreg
##       0.8084  -0.9106  -0.0212  0.7264
## s.e.  0.0158   0.0196   0.0141  0.0177
##
## sigma^2 = 8.391e-08:  log likelihood = 47500.14
## AIC=-94990.28   AICc=-94990.27   BIC=-94955.97
```

`auto.arima()` chose an $ARMA(1,2)$ model for the residuals. The $AICc$ value is -94,990.28. The fitted model is

$$(1-0.80841B)(Y_t-0.72643X_t) = (1-0.91065B-0.02121B^2)\varepsilon_t$$
$$\varepsilon_t \sim WN(0{,}8.3913 \times 10^{-8})$$

```
futures.arimax.res <- residuals(futures.auto.arimax)
checkresiduals(futures.arimax.res)
```




```
Box.test(futures.arimax.res, lag = 20, type = 'Ljung-Box')
```

```
##
##  Box-Ljung test
##
## data:  futures.arimax.res
## X-squared = 21.29, df = 20, p-value = 0.3803
```

The time series of residuals is stationary and occasionally spikes. The errors don't appear to be normally distributed. It looks like the kurtosis is too high, but the distribution is symmetric. The ACF shows a couple of significant lags, but they are late in the sequence and possibly due to sampling variability. In fact, the Ljung-Box test gives a $P-$value of 0.3803, which is not enough evidence to reject the null hypothesis that the errors are uncorrelated. All things considered, the $ARIMAX(1,0,2)$ model is a fairly good fit.

(c) `forecast()`

```r
diff.log.spot <- 0.0002
futures.fc <- forecast(futures.auto.arimax, xreg = diff.log.spot)
futures.pred.int <- data.frame(futures.fc$lower[1, 2], futures.fc$mean[1], futures.fc$upper[1, 2])
colnames(futures.pred.int) <- c('Lower Bound', '1-Step Forecast', 'Upper Bound')
rownames(futures.pred.int) <- '95% Prediction Interval'
knitr::kable(futures.pred.int, align = 'c', digits = 7)
```

	Lower Bound	1-Step Forecast	Upper Bound
95% Prediction Interval	-0.0003743	0.0001935	0.0007612

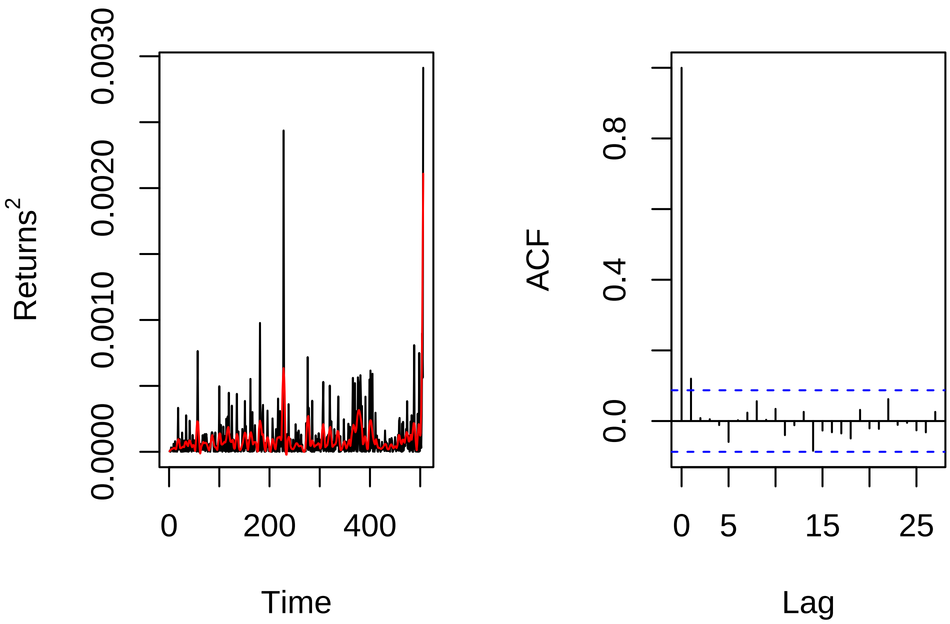
Question 2

(a) S&P 500 Log Returns

```r
data(SP500, package="Ecdat")
ret <- SP500$r500[(1804-2*253+1):1804]
ret.black.mon <- SP500$r500[1805]
par(mfrow = c(1, 2))
plot(time(ret), ret^2, type = 'l', xlab = 'Time', ylab = expression(Returns^2))
smoother <- loess(I(ret^2)~time(ret), span = 0.03)
lines(time(ret), fitted(smoother), col = 'red', lwd = 1)
ret.fit.mean <- auto.arima(ret)
ret.fit.mean
```

```
## Series: ret
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##          ma1   mean
##       0.1266  8e-04
## s.e.  0.0449  5e-04
##
## sigma^2 = 9.489e-05:  log likelihood = 1626.49
## AIC=-3246.98   AICc=-3246.93   BIC=-3234.3
```

```r
acf(ret, main = '')
```



An $ARIMA$ process is not adequate to model the returns because there is conditional heteroscedasticity, which is evident by the spikes in the plot of the squared returns versus time. The process should have an $MA(1)$ component because the ACF of returns shows a significant first lag then cuts off, and because `auto.arima()` chose an $MA(1)$ model, which means there is a high likelihood that the process models the conditional mean well. The $GARCH(1,1)$ component should be considered because it is known that the process has a high likelihood of adequately modeling the conditional variance.

(b) Fit the MA(1)+GARCH(1,1) Model

```r
ret.armagarch <- garchFit(~arma(0,1)+garch(1,1), data = ret,
                          cond.dist = 'std', trace = FALSE)
summary(ret.armagarch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 1) + garch(1, 1), data = ret, cond.dist = "std",
##     trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(0, 1) + garch(1, 1)
## <environment: 0x0000000016046c58>
##  [data = ret]
##
```

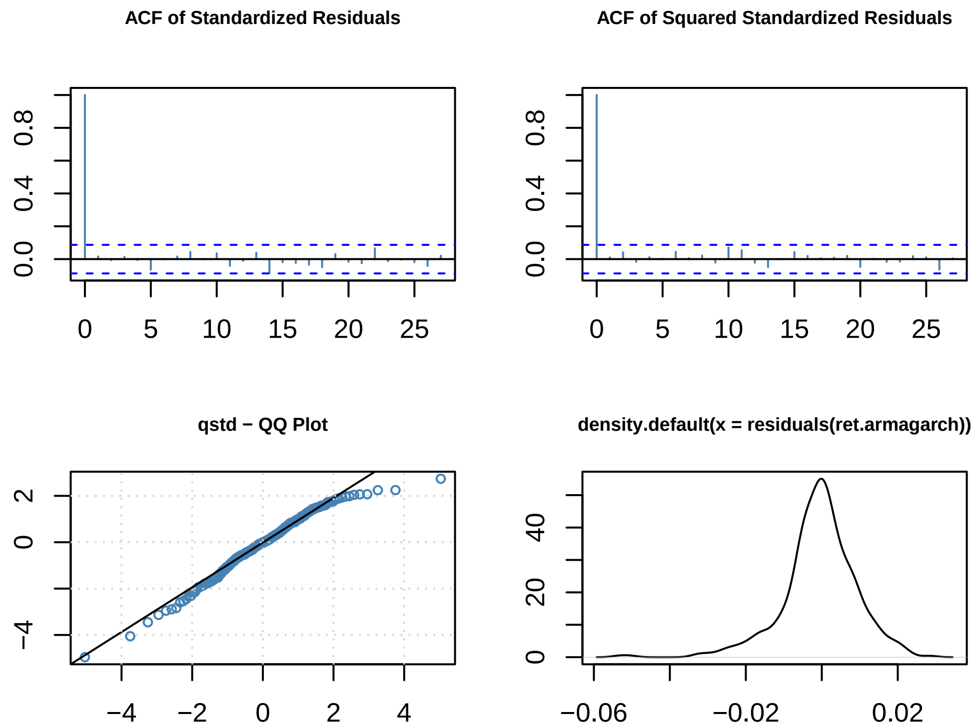
```
## Conditional Distribution:
##  std
##
## Coefficient(s):
##         mu         ma1       omega      alpha1       beta1       shape
## 0.00141651  0.09477637  0.00001075  0.04388364  0.85698004  4.08099919
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      1.417e-03   4.086e-04    3.467 0.000526 ***
## ma1     9.478e-02   4.443e-02    2.133 0.032898 *
## omega   1.075e-05   8.591e-06    1.251 0.210846
## alpha1  4.388e-02   3.283e-02    1.337 0.181384
## beta1   8.570e-01   9.709e-02    8.826  < 2e-16 ***
## shape   4.081e+00   9.239e-01    4.417 9.99e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1655.48    normalized:  3.271699
##
## Description:
##  Mon Mar 07 20:24:09 2022 by user: ZenWarrior
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  150.6831  0
##  Shapiro-Wilk Test  R    W      0.9663883 2.324834e-09
##  Ljung-Box Test     R    Q(10)  4.173439  0.9391838
##  Ljung-Box Test     R    Q(15)  9.965657  0.8218946
##  Ljung-Box Test     R    Q(20)  12.7511   0.8878061
##  Ljung-Box Test     R^2  Q(10)  5.311719  0.8694062
##  Ljung-Box Test     R^2  Q(15)  9.316155  0.860415
##  Ljung-Box Test     R^2  Q(20)  10.98605  0.9465835
##  LM Arch Test       R    TR^2   8.304252  0.7609251
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -6.519682 -6.469565 -6.519959 -6.500026
```

The fitted model is

$$\begin{aligned} Y_t - 0.00142 &= 0.09478 a_{t-1} + a_t \\ a_t &= \sigma_t \varepsilon_t,\ \{\varepsilon_t\} \sim t(4.081) \\ \sigma_t^2 &= 1.075 \times 10^{-5} + 0.04388 a_{t-1}^2 + 0.85698 \sigma_{t-1}^2 \end{aligned}$$

```r
plot.new()
par(mfrow = c(2, 2), mar = c(3, 2, 3, 2), cex.main = 0.75)
plot(ret.armagarch, which=10)
```

```r
plot(ret.armagarch, which=11)
plot(ret.armagarch, which=13)
plot(density(residuals(ret.armagarch)))
```



The Jarque-Bera and Shapiro-Wilk tests for normality are highly significant, which indicates that the standardized residuals are not normally distributed. The Ljung-Box test on the residuals fails to reject the zero correlation for lags up to 10, 15, and 20. The same is true of the squared residuals. The LM Arch Test also fails to reject the null that all ARCH coefficients of the standardized residuals are zero.

Neither the ACF of standardized residuals nor the ACF of squared standardized residuals show any significant correlations, which corroborates the Ljung-Box tests. The QQ-Plot shows some negative skew that I initially thought might be an issue, but it looks like it's due to a handful of outliers. The density plot actually looks like a t-distribution, so I don't think there is an issue with the shape of the residuals.

(c) Conditional Distribution of $\hat{\mathbf{Y}}_{\mathbf{n+1}}$

```r
# plot(density(ret))
ret.pred <- predict(ret.armagarch)
a <- ret.pred$meanForecast[1]
b <- ret.pred$standardDeviation[1]
nu <- ret.armagarch@fit[["coef"]][["shape"]]
knitr::kable(data.frame(a, b, nu), align = 'c')
```

a	b	nu
-0.0036348	0.0168394	4.080999

(d) VaR

```r
alpha <- 0.001
VaR <- qstd(alpha, mean = a, sd = b, nu = nu)
knitr::kable(data.frame(ret.black.mon, VaR), align = 'c')
```

ret.black.mon	VaR
-0.2280063	-0.0881783

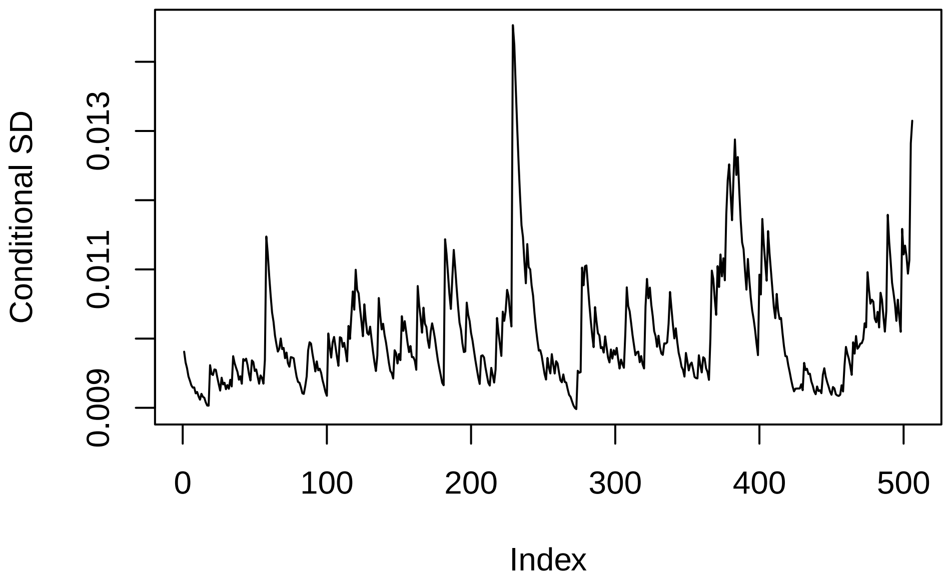
```r
ret.black.mon < VaR
```

```
## [1] TRUE
```

Yes, the loss for the log return on Black Monday far exceeded the 99.9% Value-at-risk. The market truly Fell On Black Days that Monday.

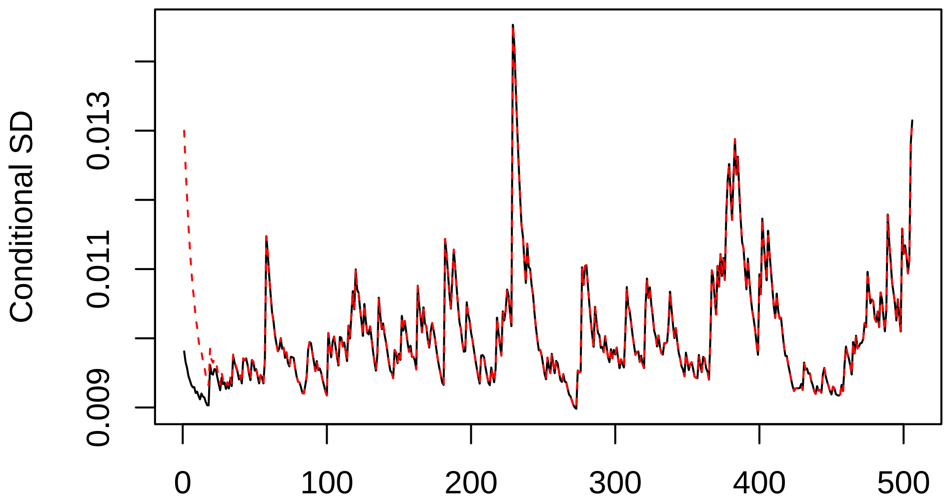
(e) Plot of Conditional Standard Deviation from the Model

```
plot(ret.armagarch@sigma.t, type = 'l', ylab = 'Conditional SD')
```



(f) Plot the Conditional Standard Deviation from Iteration

```
omega <- ret.armagarch@fit[["coef"]][["omega"]]
alpha1 <- ret.armagarch@fit[["coef"]][["alpha1"]]
beta1 <- ret.armagarch@fit[["coef"]][["beta1"]]
next.sigma <- function(a, sigma) {
  return (sqrt(omega + alpha1 * a ^ 2 + beta1 * sigma ^ 2))
}
sigma.array <- array(NA, dim = length(ret))
sigma.array[1] <- 0.013
a.past <- residuals(ret.armagarch, standardize = FALSE)
for (j in 2:length(ret)) {
  sigma.array[j] <- next.sigma(a.past[j-1], sigma.array[j-1])
}
plot.new()
plot(time(ret), ret.armagarch@sigma.t, type = 'l',
     xlab = '', ylab = 'Conditional SD')
lines(time(ret), sigma.array, type = 'l', col = 'red', lty = 2)
```



(g) Convergence Induced by the GARCH Formula

Here we have $\alpha + \beta = 0.90086 < 1$, so $\{a_t\}$ is stationary and it has a marginal variance equal to $\frac{\omega}{1-\alpha-\beta}$. At time $t = 1$, even though the initial values of the series as so disparate, they are both expected to converge to this marginal variance as the number of time steps increases. The GARCH formula shows that the conditional variance contains information about previous values of the variance, but the most recent is by far the most influential, so the influence of the exceptionally high value of 0.013 dissipates over time.

$$\begin{aligned}
\sigma_3^2 &= \omega + \alpha a_2^2 + \beta\sigma_2^2 \\
&= \omega + \alpha\sigma_2^2\varepsilon_2^2 + \beta\sigma_2^2 \\
&= \omega + \left(\alpha\varepsilon_2^2 + \beta\right)\left(\omega + \alpha a_1^2 + \beta\sigma_1^2\right) \\
&= \omega + \left(\alpha\varepsilon_2^2 + \beta\right)\left[\omega + \alpha\sigma_1^2\varepsilon_1^2 + \beta\sigma_1^2\right] \\
&= \omega + \left(\alpha\varepsilon_2^2 + \beta\right)\left[\omega + \left(\alpha\varepsilon_1^2 + \beta\right)\sigma_1^2\right] \\
&= \omega\left(1 + \alpha\varepsilon_2^2 + \beta\right) + \left(\alpha\varepsilon_2^2 + \beta\right)\left(\alpha\varepsilon_1^2 + \beta\right)\sigma_1^2
\end{aligned}$$

After two steps from $t = 1$, the influence of σ_1^2 is reduced by a factor of $\left(\alpha\varepsilon_2^2 + \beta\right)\left(\alpha\varepsilon_1^2 + \beta\right)$. After h steps, the influence of σ_1^2 is reduced by a factor of $\prod_{j=1}^{h}\left(\alpha\varepsilon_j^2 + \beta\right)$. Because α and β are small, this product should dampen the influence of σ_1^2 even in the presence of some occasionally large values of ε_j. The process eventually all but forgets about σ_1^2 and moves on with its life.

Question 3

Average Predicted Variance

Use the formula $E[\sigma^2_{n+h}|\mathcal{F}_{n-1}] = \omega\frac{1-(\alpha+\beta)^h}{1-(\alpha+\beta)} + (\alpha+\beta)^h\sigma^2_n = v[1-e^{-ah}] + e^{-ah}\sigma^2_n$ for $h \in \{0, 1, ..., T-1\}$, $v = \frac{\omega}{1-\alpha-\beta}$, and $(\alpha+\beta) = e^{-a}$.

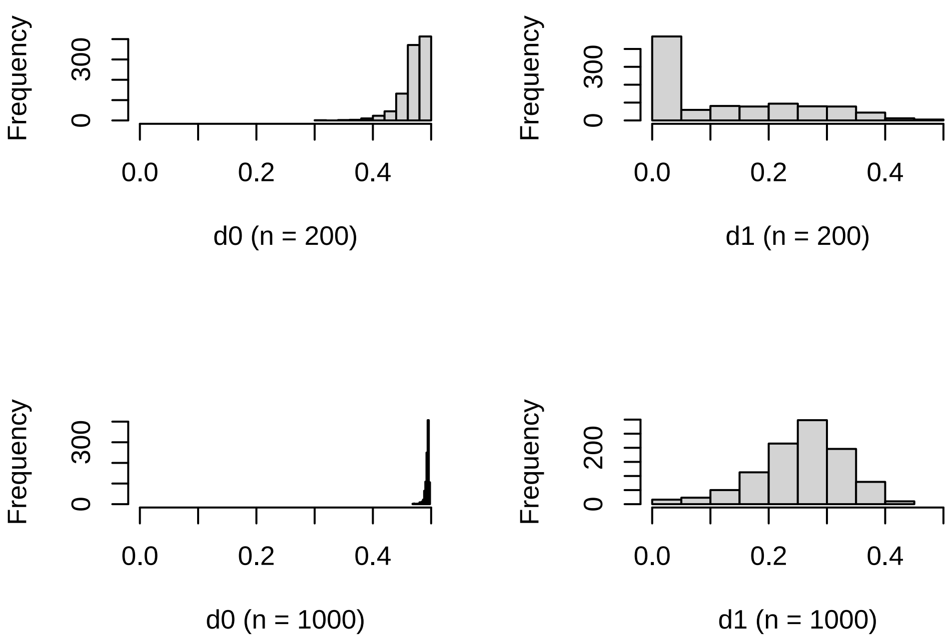
$$
\begin{aligned}
\bar{\sigma}_{n-1}(T) &= \frac{1}{T}\sum_{h=0}^{T-1}\{v - v\cdot e^{-a\cdot h} + e^{-a\cdot h}\sigma^2_n\} && (1)\\
&= v + \frac{\left(\sigma^2_n - v\right)}{T}\sum_{h=0}^{T-1} e^{-a\cdot h} && (2)\\
&= v + \frac{\left(\sigma^2_n - v\right)}{T}\cdot\frac{(1)\left(1-e^{-a\cdot T}\right)}{\left(1-e^{-a}\right)} && (3)\\
&= v + \frac{\left(\sigma^2_n - v\right)\left(1-e^{-a\cdot T}\right)}{T\left(1-e^{-a}\right)} && (4)
\end{aligned}
$$

At step (2), use the fact that $(\sigma^2_n - v)$ is known at time $n-1$. Step (3) simply uses the formula for a geometric partial sum where $r = e^{-a}$.

Question 4

ARFIMA Simulation

```
set.seed(343)
sentinel <- 1000
phi <- 0.35
sigma2 <- 1
n <- c(200, 1000)
par(mfrow = c(2, 2))
for(j in 1:2) {
  d0 <- array(NA, n[j])
  d1 <- array(NA, n[j])
  for(i in 1:sentinel) {
    fd.sim <- fracdiff.sim(n[j], ar = phi, d = 0.3)$series
    # plot(fd.sim, type = 'l', xlab = 't', ylab = '$ARFIMA(0,0.3,0)$')
    # acf(fd.sim)
    d0[i] <- fracdiff(fd.sim, nar = 0, nma = 0)$d
    d1[i] <- fracdiff(fd.sim, nar = 1, nma = 1)$d
  }
  hist(d0, xlab = paste0('d0', ' (n = ', n[j], ')'), xlim = c(0, 0.5), main = '')
  hist(d1, xlab = paste0('d1', ' (n = ', n[j], ')'), xlim = c(0, 0.5), main = '')
}
```



There is a bias-variance tradeoff between the $ARFIMA(0, d, 0)$ model and the $ARFIMA(1, d, 1)$ model. The $ARFIMA(0, d, 0)$ model is biased towards some value near the upper bound of 0.5 and has less variability in the sampling distribution. The $ARFIMA(1, d, 1)$ is less biased and has more variability. As n increases, the

$ARFIMA(0, d, 0)$ model becomes less variable, while the $ARFIMA(1, d, 1)$ model becomes less biased. The shape of the $ARFIMA(1, d, 1)$ model is becoming more symmetric and the center seems to be approaching the population difference of $d = 0.3$.