Empirical Methods Homework 6

Group 9: Linqi Huang, Abhesh Kumar, Yu Onohara, Maitrayee Patil, Redmond Xia

Problem1

1.

```
# Impoting and organizing the data
ddf = read.csv("F-F_Research_Data_5_Factors_2x3_daily.CSV", skip = 3 ,stringsAsFactors=FALSE)
mdf = read.csv("F-F_Research_Data_5_Factors_2x3.CSV", skip = 3, stringsAsFactors=FALSE)
mdf = mdf[1:678,]
ddf$X = as.Date(as.character(ddf$X), "%Y%m%d")
mdf$X = as.numeric(mdf$X)*100 + 1
mdf$X = as.Date(as.character(mdf$X), "%Y%m%d")
dCMA = as.ts(as.numeric(ddf$CMA))
mCMA = as.ts(as.numeric(mdf$CMA))

## Problem 1-1
# Run the ARMA(1,1) model
mout1 = arima(mCMA, c(1,0,1))
mout1
```

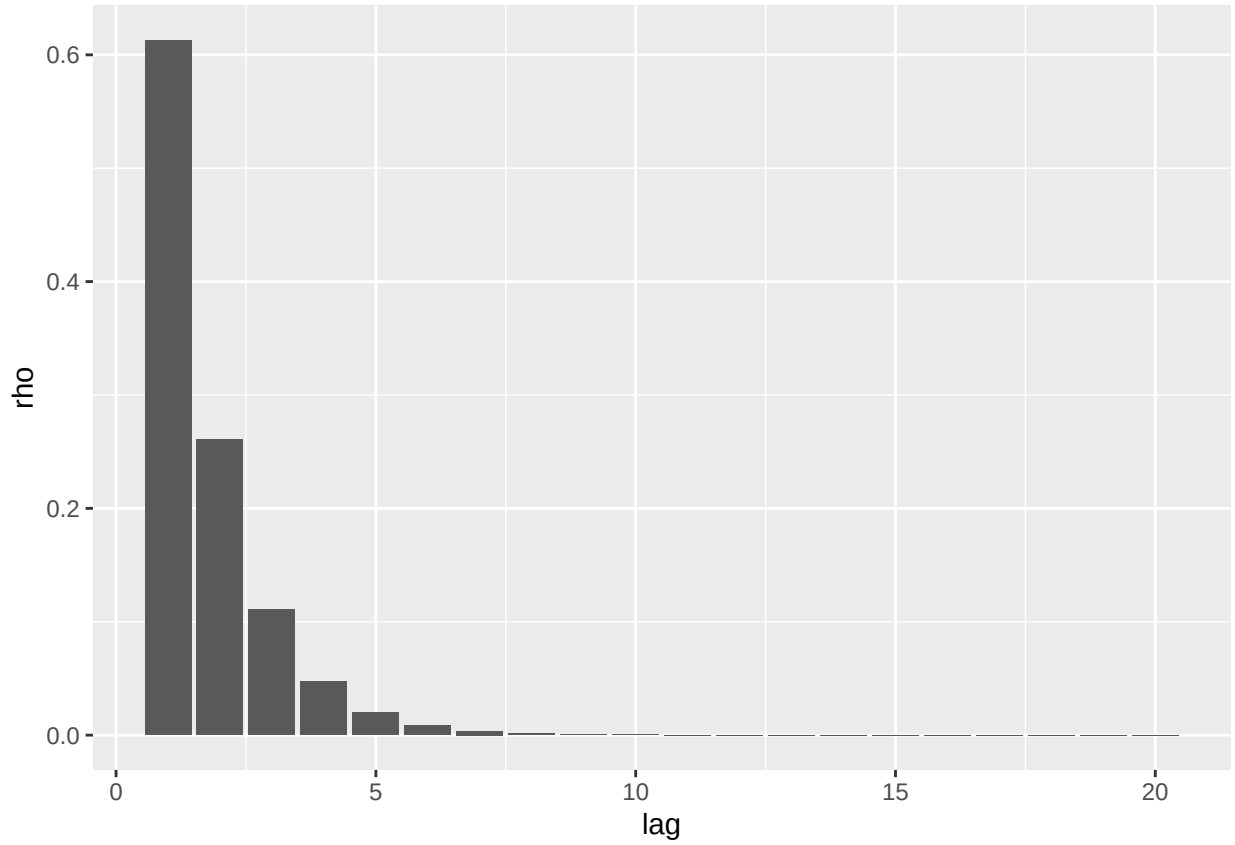
```
##
## Call:
## arima(x = mCMA, order = c(1, 0, 1))
##
## Coefficients:
##          ar1      ma1  intercept
##       0.4252  -0.3120     0.2729
## s.e.  0.2525   0.2652     0.0908
##
## sigma^2 estimated as 3.901:  log likelihood = -1423.55,  aic = 2855.09
```

```
# Store the parameters
phi1 = as.numeric(mout1$coef[1])
theta1 = as.numeric(mout1$coef[2])
sigma = sqrt(var(as.numeric(mout1$residuals)))

# Caluculating the covariance with lag 0 (= Variance of yt)
gamma0 = sigma^2 * (1 + theta1^2 - 2 * phi1 * theta1) / (1 - phi1^2)
gamma0
```

```
## [1] 6.499611
```

```
# Using gamma above, compute the first order autocorrelation
rho = double()
rho[1] = phi1 - theta1 * (sigma^2 / gamma0)
for(i in 2:20){
  rho[i] = rho[i-1] * phi1
}
rho = as.data.frame(rho)
rho$lag = seq(1,20,1)
plot1 = ggplot(rho, aes(x = lag, y = rho)) + geom_bar(stat = "identity")
```

```
plot1
```




```
# Calculating the half life
mhl = log(0.5)/log(as.numeric(mout1$coef[1]))
mhl
```

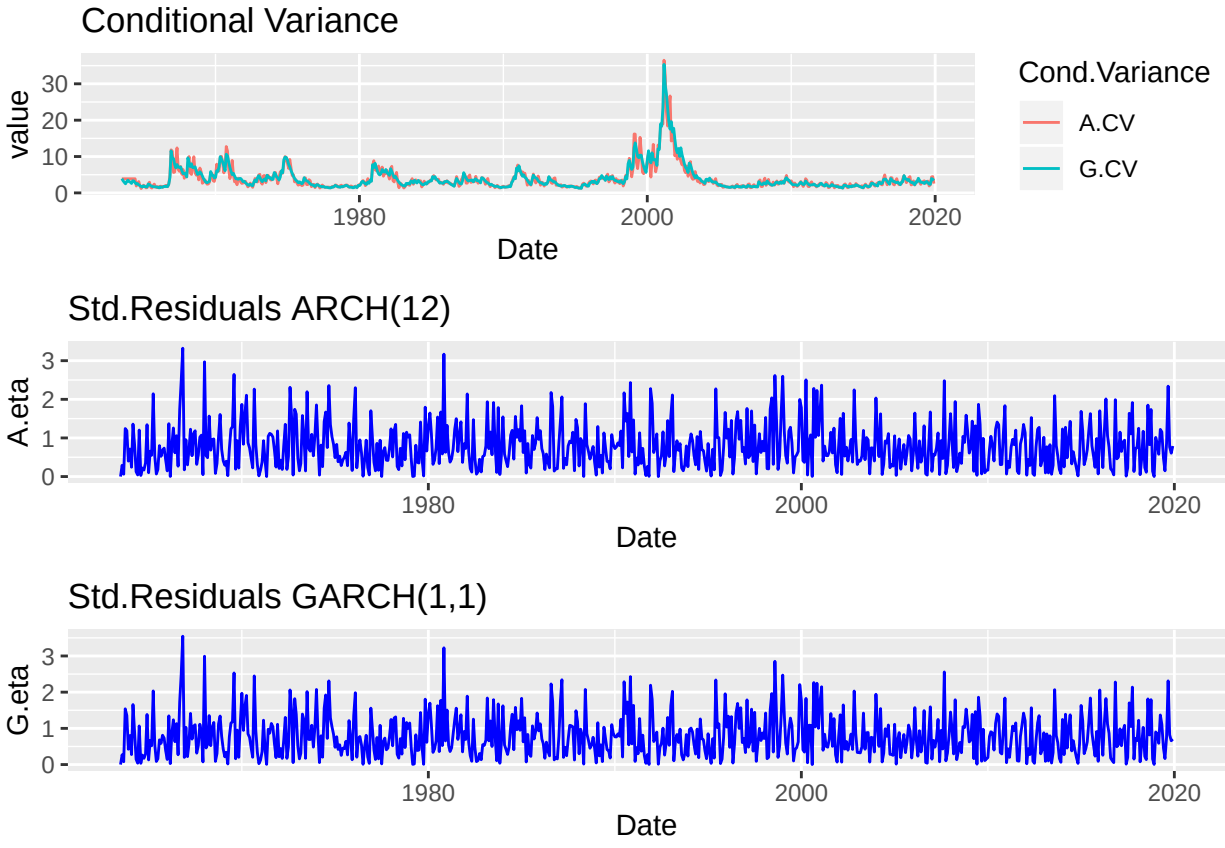
```
## [1] 0.8105551
```

2 and 3.

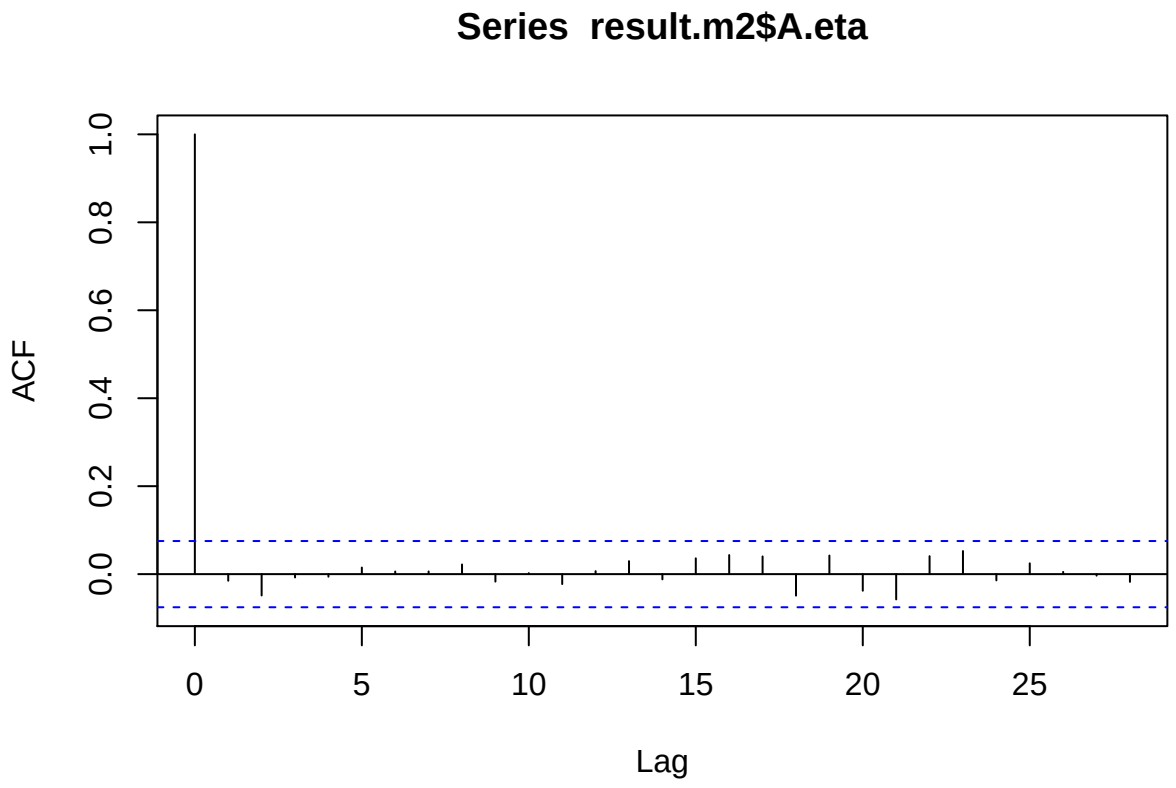
Since we can find time-varying volatility, these estimated variance processes are not stationary. Additionally, the standadized volatilities seem stationary, so these models are effective for accounting for clustering of volatility.

```
## Problem 1-2 and 1-3
# Run the arch(12) and garch(1,1) model
m.arch.12 = garchFit(formula = ~ arma(1,1) + garch(12,0), data = mCMA, trace = F)
m.garch.1.1 = garchFit(formula = ~ arma(1,1) + garch(1,1), data = mCMA, trace = F)

# Summarizing the results
result.m1 = data.frame(matrix(ncol = 0, nrow = length(mdf$X))) %>%
  mutate(Date = mdf$X) %>%
  mutate(A.CV = m.arch.12@h.t) %>%
  mutate(G.CV = m.garch.1.1@h.t)

result.m2 = data.frame(matrix(ncol = 0, nrow = length(mdf$X))) %>%
```

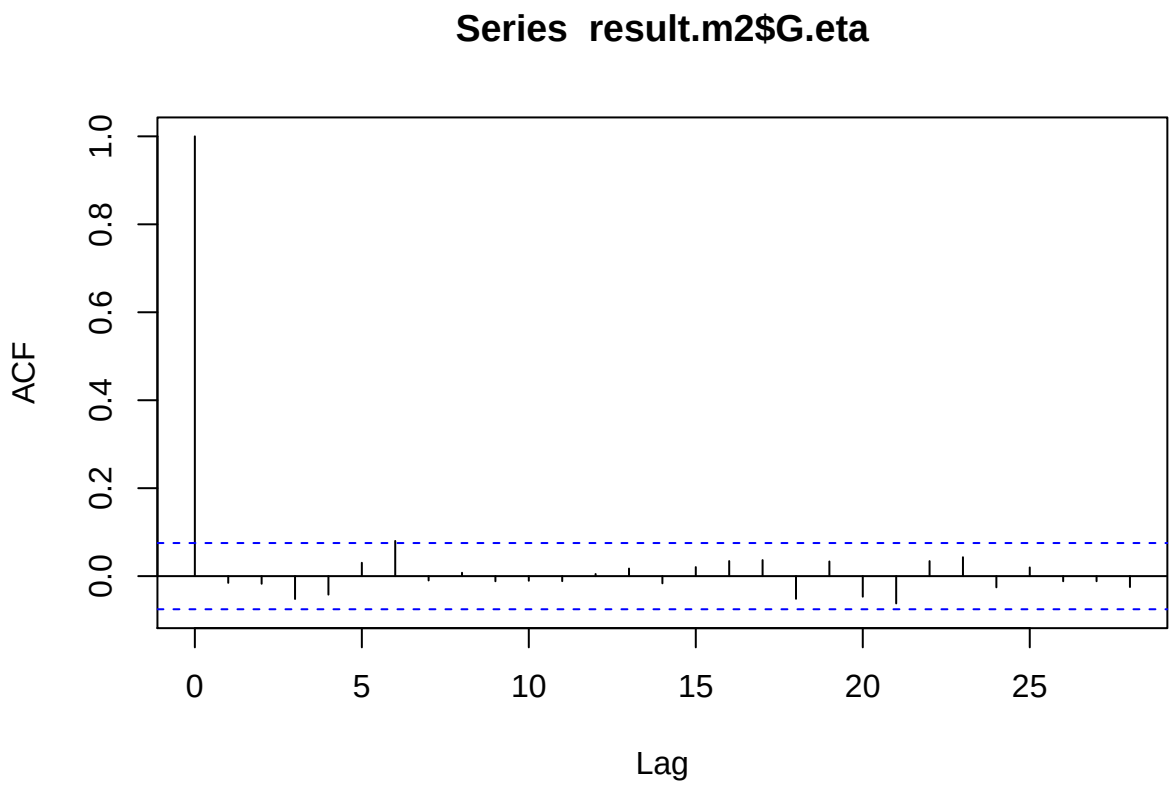
```
  mutate(Date = mdf$X) %>%
  mutate(A.eta = abs(residuals(m.arch.12, standardize = T))) %>%
  mutate(G.eta = abs(residuals(m.garch.1.1, standardize = T)))

result.m1long = gather(result.m1, key = "Cond.Variance", value = value, -Date)
result.m2long = gather(result.m2, key = "Std. Residuals", value = value, -Date)

# Plot the conditional variance and the standadized residuals
m1.base = ggplot(data = result.m1long, aes(x = Date, y = value, colour = Cond.Variance))
plot2 = m1.base + geom_line() + ggtitle("Conditional Variance")
plot3 = ggplot(data = result.m2, aes(x = Date, y = A.eta)) + geom_line(colour = "blue") + ggtitle("Std.
plot4 = ggplot(data = result.m2, aes(x = Date, y = G.eta)) + geom_line(colour = "blue") + ggtitle("Std.
multiplot(plot2,plot3,plot4)
```




```
acf(result.m2$A.eta)
```

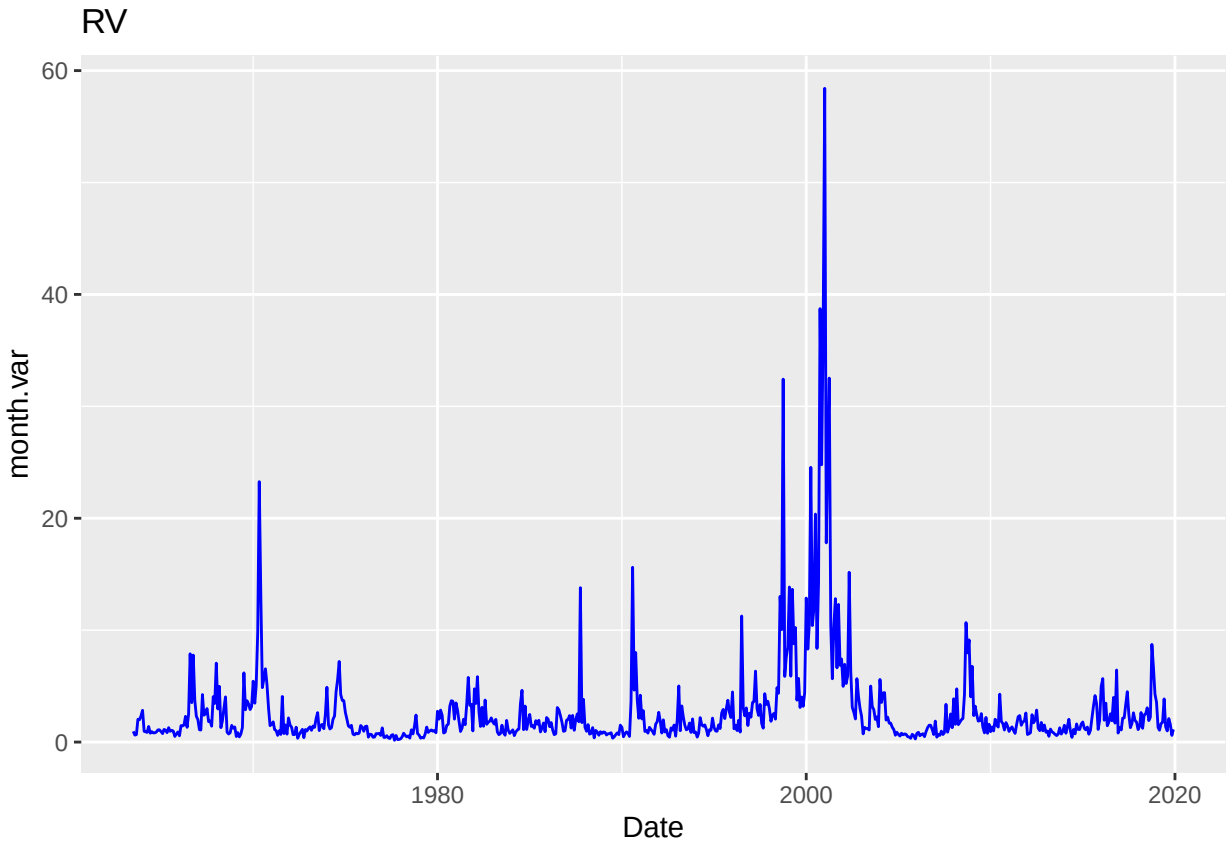



```
acf(result.m2$G.eta)
```



Problem 1-4

```
## Problem 1-4
# Subset monthly data and calculate monthly variance
ddf.xts = xts(ddf, order.by = ddf$X)
month.end = endpoints(ddf.xts, on = "month")
month.beg = head(endpoints(ddf.xts, on='month')+1, -1)
month.var = double(length(month.beg))
ddf$sq.ret = ddf$CMA^2
for (i in (1:length(month.beg))){
  month.var[i] = sum(ddf$sq.ret[month.beg[i]:month.end[i+1]])
}

# Store the value and plot the graph
result.d = data.frame(matrix(ncol = 0, nrow = length(month.var))) %>%
  mutate(Date = mdf$X) %>%
  mutate(month.Var = month.var)

d.base = ggplot(data = result.d, aes(x = Date))
plot5 = d.base + geom_line(aes(y = month.var), colour = "blue") + ggtitle("RV")
plot5
```



Problem 1-5

```
## Problem 1-5
# Compute the first order autocorrelations
acf.arch.1st = acf(m.arch.12@residuals, plot = F)[1]
acf.arch.1st = as.numeric(acf.arch.1st$acf)
acf.garch.1st = acf(m.garch.1.1@residuals, plot = F)[1]
acf.garch.1st = as.numeric(acf.garch.1st$acf)
acf.RV.1st = acf(result.d$month.Var, plot = F)[1]
acf.RV.1st = as.numeric(acf.RV.1st$acf)
corr = c(acf.arch.1st, acf.garch.1st, acf.RV.1st)
corr
```

```
## [1] -0.0136731 -0.0264952  0.6910204
```

```
# Compute the correlations
corr.RV_archRes = cor(m.arch.12@residuals, result.d$month.Var)
corr.RV_garchRes = cor(m.garch.1.1@residuals, result.d$month.Var)
corr.RV_garchFitted = cor(m.garch.1.1@h.t, result.d$month.Var)
corr.garchRes_garchFitted = cor(m.garch.1.1@residuals, m.garch.1.1@h.t)
corr.RV_archRes
```

```
## [1] 0.1620371
```

```
corr.RV_garchRes
```

```
## [1] 0.1580956
```

```
corr.RV_garchFitted
```

```
## [1] 0.6319828
```

```
corr.garchRes_garchFitted
```

```
## [1] 0.11976
```

Problem 1-6

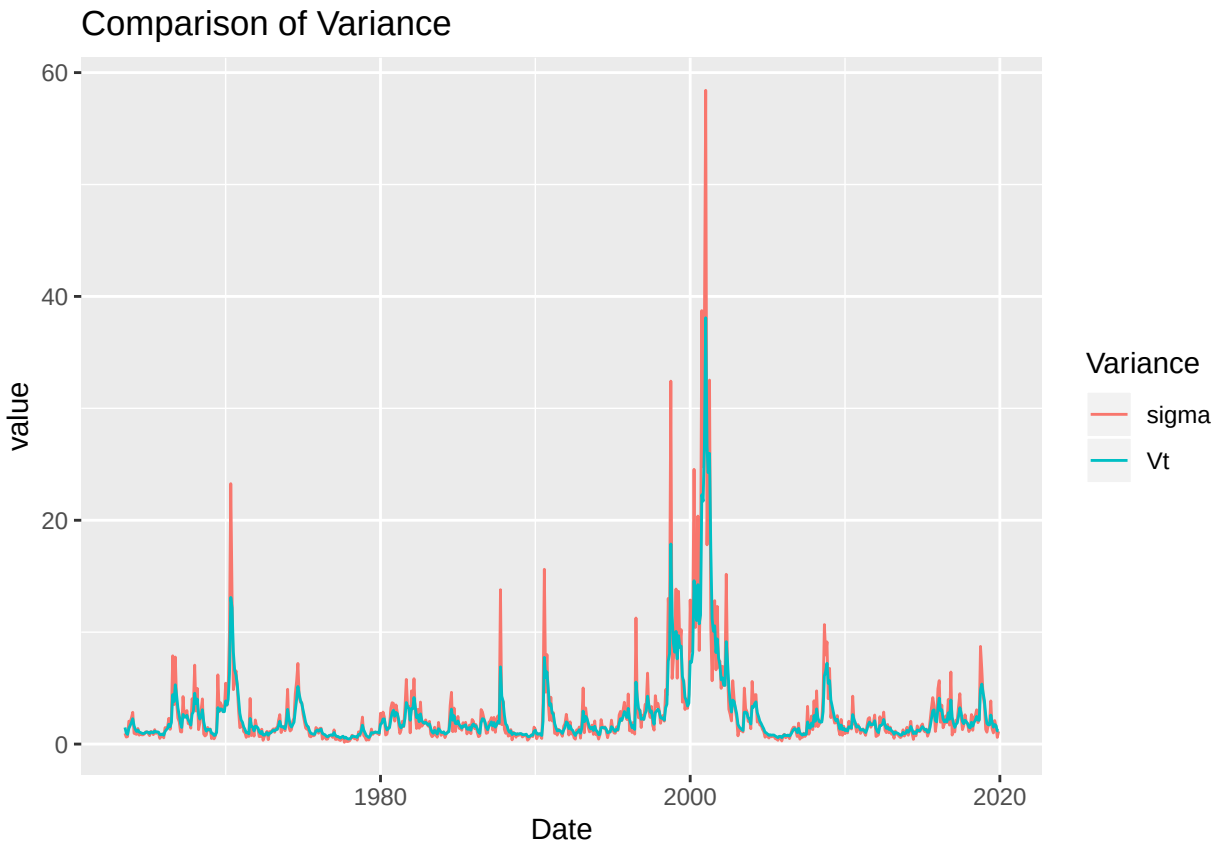
```
## Problem 1-6
# Run the arma(1,1) model
RV_arma1.1 = arima(result.d$month.Var, order = c(1,0,1))
RV_arma1.1
```

```
##
## Call:
## arima(x = result.d$month.Var, order = c(1, 0, 1))
##
## Coefficients:
##          ar1      ma1  intercept
##       0.9220  -0.4977     2.7396
## s.e.  0.0181   0.0388     0.7366
##
## sigma^2 estimated as 9.154:  log likelihood = -1713.13,  aic = 3434.25
```

```
par = as.numeric(RV_arma1.1$coef)
epsilon = RV_arma1.1$residuals
mu = par[3]/(1 - par[1])

# Compute the time series of vt
vt = double(length(result.d$month.Var-1))
for(j in 1:length(vt)){
  vt[j] = par[1]*result.d$month.Var[j] + par[2]*epsilon[j]
}

# Compute correlation and plot the result
corr.RV_arma = cor(result.d$month.Var, vt)
corr.RV_arma
```

```
## [1] 0.9425451
```

```
result.arma = data.frame(matrix(ncol = 0, nrow = length(mdf$X))) %>%
  mutate(Date = mdf$X) %>%
  mutate(Vt = vt)%>%
  mutate(sigma = result.d$month.Var)
result.arma.long = gather(result.arma, key = "Variance", value = value, -Date)

d.base = ggplot(data = result.arma.long, aes(x = Date, y = value, colour = Variance))
plot6 = d.base + geom_line() + ggtitle("Comparison of Variance")
plot6
```


Comparison of Variance
60
40
20
0
value
1980
2000
2020
Date
Variance
sigma
Vt