# Problem Set 4, Econ 211C

Swati Sharma

## Question 1 (40 points)

Consider a special case of the long-run risks model:

$$x_{t+1} = \rho x_t + \varphi_e \sigma e_{t+1} \tag{1}$$
$$g_{t+1} = \log\left(C_{t+1}/C_t\right) = \mu + x_t + \sigma \eta_{t+1} \tag{2}$$
$$e_{t+1}, \eta_{t+1} \overset{i.i.d.}{\sim} N(0,1) \tag{3}$$
$$\tag{4}$$

$\mu = 0.0015$, $\sigma = 0.0078$, $\rho = 0.979$, $\varphi_e = 0.044$, and $\varphi_d = 4.5$. Download real personal consumption expenditures data from the BEA for the period Jan 1999 – Mar 2018 and use the sum of non-durable and services as a monthly series for consumption (note: the Quandl code for the data is `BEA/T20806_M`). Estimate values for the latent state $\hat{\xi}_{t+1|t} = \hat{x}_{t+1|t}$ for each month using the Kalman filter. Write the recursions yourself (without using any `R` packages) and plot the forecasts of the latent states.

**Solution:**

```
consumption = Quandl("BEA/T20806_M",start_date="1999-01-01",end_date="2018-03-01",type="xts")
temp = as.data.frame(consumption)
```

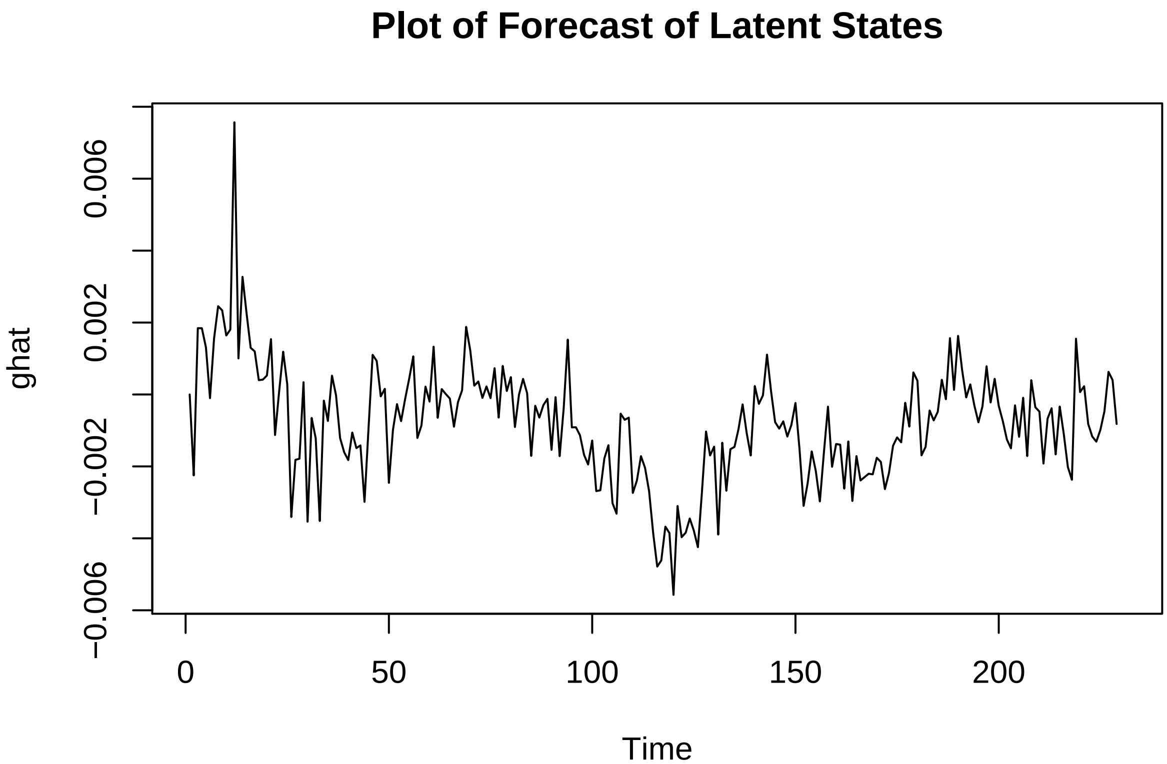
```
temp$consumption = temp$`:Nondurable goods` + temp$Services

temp <- slide(temp, Var = "consumption", slideBy = -1)
```

```
##
## Remember to put temp in time order before running.

##
## Lagging consumption by 1 time units.
```

```
temp$g_t = log(temp$consumption/temp$`consumption-1`)

temp <- temp[,-c(1:10)]

g_t = temp$g_t
```

```
xhat = rep(0, nrow(temp))
ghat = rep(0, nrow(temp))
P = rep(0, nrow(temp))

epsilon <- rnorm(nrow(temp),0,1)
eta <- rnorm(nrow(temp),0,1)
```

```r
mu = 0.0015
sigma = 0.0078
rho = 0.979
phi_e = 0.044
phi_d = 4.5

# initializing guess
xhat_init <- 0
P_init <- 1
H <- 1
A <- mu
R <- sigma^2
x_t <- 1
F <- rho
Q <- (phi_e^2)*(sigma^2)
r = sigma*eta*phi_e
u = sigma*eta


S = rep(0, nrow(temp))
S[1] = H*P_init*H+R

xhat[1] = xhat_init + r[2]+ P_init*H*(S[1])*(g_t[2]-A*x_t+H*xhat_init-u[2])
P[1]= P_init - P_init*H*(S[1])*H*P_init

for(i in 1: nrow(temp)) {
xhat[i+1] = F*xhat_init+ r[i+1]+ F*P_init*H*(S[i])*(g_t[i+1]-A*x_t+H*xhat[i]-u[2])
P[i+1] = F*(P_init-P_init*H*(S[i])*H*P_init)*F+Q

ghat[i+1] = A*x_t+H*xhat[i+1]+r[i+2]
S[i+1] = H*P[i+1]*H+R
}

plot(ghat, main = "Plot of Forecast of Latent States", type = "l", xlab = "Time")
```




## Question 2 (30 points)

Suppose that $r_1$, $r_2$, $\ldots$, $r_n$ are observations of a return series that follows the $AR(1)-GARCH(1,1)$ model:

$$r_t = \mu + \phi r_{t-1} + \sigma_t \varepsilon_t \tag{5}$$
$$\sigma_t^2 = \alpha_0 + \alpha_1 \sigma_{t-1}^2 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \tag{6}$$

where $\varepsilon_t \overset{i.i.d.}{\sim} N(0,1)$. Derive the conditional log likelihood of the data.

**Solution:**

Given the above equations we can determine that

$$E(r_t) = \mu + \phi_{t-1}$$
$$ML(\theta|y_1) = \prod f_{Y_t|Y_{t-1}}$$

where

$$f_{Y_t|Y_{t-1}} = N(\mu + \phi_{t-1}, \alpha_0 + \alpha_1 \sigma_{t-1}^2 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2)$$
$$ML(\theta|y_1) = \prod \frac{1}{\sqrt{2\pi * (\alpha_0 + \alpha_1 \sigma_{t-1}^2 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2)}} * exp(\frac{-(r_t - \mu - \phi r_{t-1})^2}{2 * (\alpha_0 + \alpha_1 \sigma_{t-1}^2 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2))})$$
$$ML(\theta|y_t) = \sum_{t=2}^{T} log(f_{Y_t|Y_{t-1}})$$
$$ML(\theta|y_t) = \sum_{t=2}^{T} = -\frac{T-1}{2} log(2\pi) - \frac{1}{2} \sum_{t=2}^{T} log(\alpha_0 + \alpha_1 \sigma_{t-1}^2 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2) - \sum_{t=2}^{T} \frac{(r_t - \mu - \phi r_{t-1})^2}{2 * (\alpha_0 + \alpha_1 \sigma_{t-1}^2 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2))}$$

# Question 3 (30 points)

Consider the monthly returns of Intel stock from January 1973 to May 2018. Transform the returns into log returns. Build a GARCH model for the transformed series and compute 1-step to 5-step ahead volatility forecasts at the forecast origin May 2018. Plot your volatility estimates for the entire period, along with your forecasts.

**Solution:**

```
library(quantmod)
```

```
## Loading required package: TTR
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:xts':
##
##     first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
getSymbols("INTC", from = "1973-01-01", to = "2018-05-31")
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
##
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).
```

```
## [1] "INTC"
```

```
monthRets <- as.data.frame(monthlyReturn(log(INTC$INTC.Adjusted)))
monthRets <- tibble::rownames_to_column(monthRets)
```

```
library(fGarch)
```

```
## Loading required package: timeDate
```

```
## Loading required package: timeSeries
```

```
##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':
##
##     time<-

## Loading required package: fBasics

##
## Attaching package: 'fBasics'

## The following object is masked from 'package:TTR':
##
##     volatility
```

```
library(ggplot2)
library(lubridate)
```

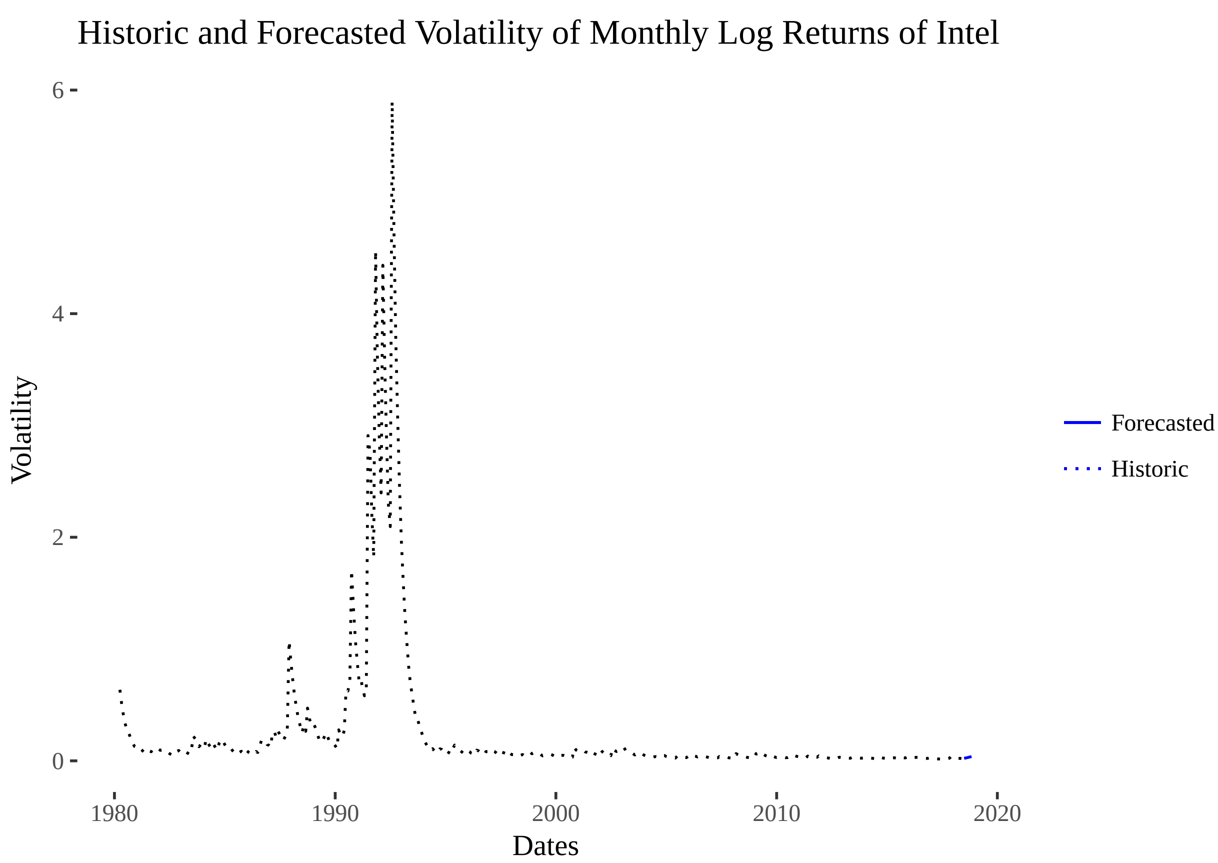
```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
library(ggthemes)
## Finding step 1:5 Volatility forecasts
garch.1 <- garchFit(formula = ~garch(1,1), data = monthRets$monthly.returns, trace = FALSE)
```

```
## Warning in sqrt(diag(fit$cvar)): NaNs produced
```

```
volGARCH <- as.data.frame(predict(garch.1, 5)$standardDeviation)
volGARCH$dates <- c("2018-06-30", "2018-07-31", "2018-08-31", "2018-09-29", "2018-10-31")

## Pulling estimated historic volatility
volMonthlyRets <- as.data.frame(volatility(garch.1,type = "sigma"))
volMonthlyRets$dates <- monthRets$rowname
```

```
ggplot()+
  geom_line(data=volMonthlyRets, aes(y=`volatility(garch.1, type = "sigma")`, x=as.Date(dates), linetype
  geom_line(data=volGARCH, aes(y=`predict(garch.1, 5)$standardDeviation`, x=as.Date(dates), linetype="F
  scale_linetype_manual(name="", values = c("Historic"=3, "Forecasted"=1))+
  ylab("Volatility")+
  xlab("Dates")+
  ggtitle("Historic and Forecasted Volatility of Monthly Log Returns of Intel")+
  theme_tufte()
```


Historic and Forecasted Volatility of Monthly Log Returns of Intel
6
4
2
0
Volatility
1980
1990
2000
2010
2020
Dates
Forecasted
Historic