Johnson & Johnson quarterly earnings per share

In [ ]:
```
# Time plot for Johnson&Johnson
plot(JohnsonJohnson, main='Johnson&Johnosn earnings per share', col='blue',
```

In [ ]:
```
# log-return of Johnson&Johnson
jj.log.return=diff(log(JohnsonJohnson))
jj.log.return.mean.zero=jj.log.return-mean(jj.log.return)
```

In [ ]:
```
# Plots for log-returns
par(mfrow=c(3,1))
plot(jj.log.return.mean.zero, main='Log-return (mean zero) of Johnson&Johnos
acf(jj.log.return.mean.zero, main='ACF')
pacf(jj.log.return.mean.zero, main='PACF')
```

In [ ]:
```
# Order
p=4
```

In [ ]:
```
# sample autocorreleation function r
r=NULL
r[1:p]=acf(jj.log.return.mean.zero, plot=F)$acf[2:(p+1)]
r
```

In [ ]:
```
# matrix R
R=matrix(1,p,p) # matrix of dimension 4 by 4, with entries all 1's.

# define non-diagonal entires of R
for(i in 1:p){
    for(j in 1:p){
        if(i!=j)
            R[i,j]=r[abs(i-j)]
        }
    }
R
```

In [ ]:
```
# b-column vector on the right
b=matrix(r,p,1)# b- column vector with no entries
b
```

In [ ]:
```
phi.hat=solve(R,b)[,1]
phi.hat
```

In [ ]:
```
# Variance estimation using Yule-Walker Estimator
c0=acf(jj.log.return.mean.zero, type='covariance', plot=F)$acf[1]
c0
var.hat=c0*(1-sum(phi.hat*r))
var.hat
```

In [ ]:

```
# Constant term in the model
phi0.hat=mean(jj.log.return)*(1-sum(phi.hat))
phi0.hat
```

In [ ]:

```
cat("Constant:", phi0.hat," Coeffcinets:", phi.hat, " and Variance:", var.ha
```

In [ ]:

```
```