Time Series Analysis

Anisha Mittal

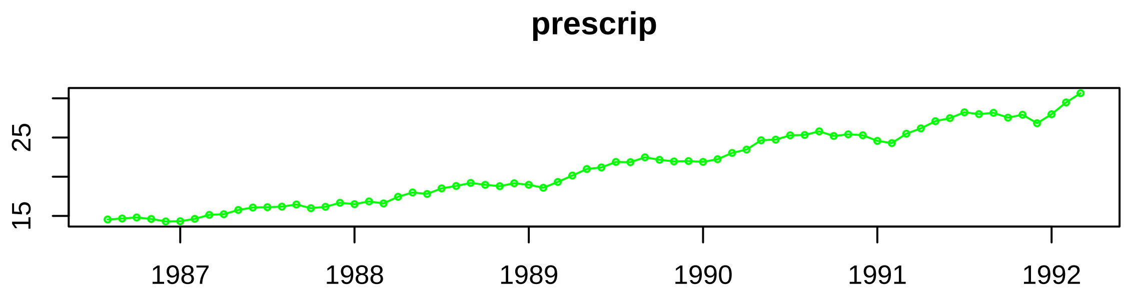
2020-12-4

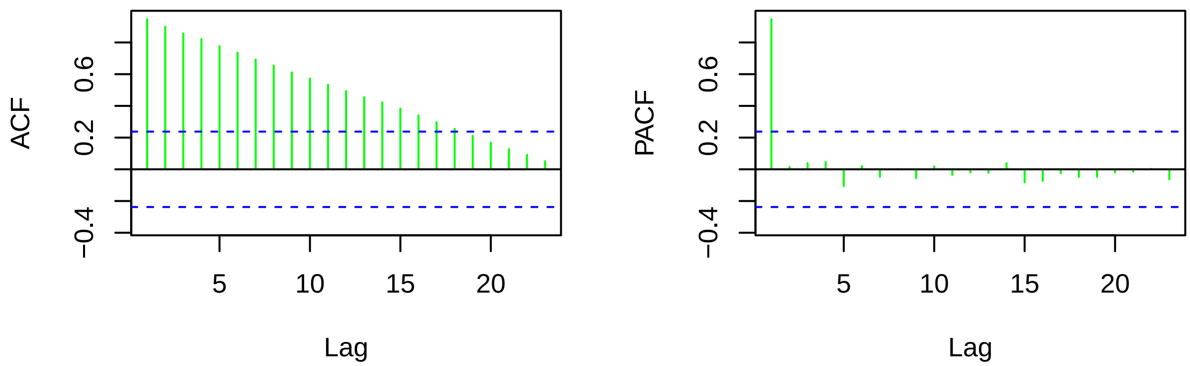
```
library(TSA)
library(forecast)
library(lmtest)
library(fGarch)
library(urca)
library(tseries)
```

1. Load Prescriptions data (already available on TSA package)

a. Augmented Diskey - Fuller test

```
data(prescrip)
tsdisplay(prescrip, col = "green")
```






```
adf.test(prescrip, k=0)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  prescrip
## Dickey-Fuller = -2.7968, Lag order = 0, p-value = 0.2515
## alternative hypothesis: stationary
```

Looking at the Augmented Dickey Fuller test, we see that the p-value $= 0.2515 > 0.05$ (5% significance level) Therefore there does not exist enough evidence to reject the null hypothesis that $\omega = 0$. That is; there is a unit root and the series need to be further differenced to get to stationarity (Alternative Hypothesis) The data plot shows a linear trend. The ACF plot decays to zero at a slow rate which also means that there exists a unit root. The ACF plots does not show any signs of seasonal component.

b. phi(3) test statistics

```
n=length(prescrip)
tt=2:n # convenience vector of time indices
y=diff(prescrip) # first difference of the series
fit=lm(y~tt+prescrip[-n]) # estimate alpha, omega x[t-1], beta
yhat=fitted(fit)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ tt + prescrip[-n])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.44289 -0.33147  0.01302  0.33358  1.04300
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.12039    1.08902   2.865  0.00563 **
## tt            0.05913    0.01987   2.975  0.00413 **
## prescrip[-n] -0.23663    0.08461  -2.797  0.00681 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4816 on 64 degrees of freedom
## Multiple R-squared:  0.1319, Adjusted R-squared:  0.1048
## F-statistic: 4.862 on 2 and 64 DF,  p-value: 0.01082
```

```r
mean(prescrip)
```

```
## [1] 21.0586
```

```r
# degrees of freedom of the model = No. of parameters - 1
SSM<-(sum((yhat-mean(y))^2))/2
# degree of freedom of the residuals = No of data points - No. of parameters
SSE<-(sum((y-yhat)^2))/64

Phi3<-(SSM)/(SSE)
Phi3
```

```
## [1] 4.861987
```

c.

```r
A<-ur.df(y,type='trend',lags=0)
summary(A)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.45664 -0.37287  0.03616  0.36303  1.03723
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.086679   0.128232   0.676    0.502
## z.lag.1     -0.886861   0.127888  -6.935  2.6e-09 ***
## tt           0.003881   0.003325   1.167    0.248
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5111 on 63 degrees of freedom
## Multiple R-squared:  0.4335, Adjusted R-squared:  0.4155
## F-statistic: 24.11 on 2 and 63 DF,  p-value: 1.68e-08
##
##
## Value of test-statistic is: -6.9347 16.0928 24.1066
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

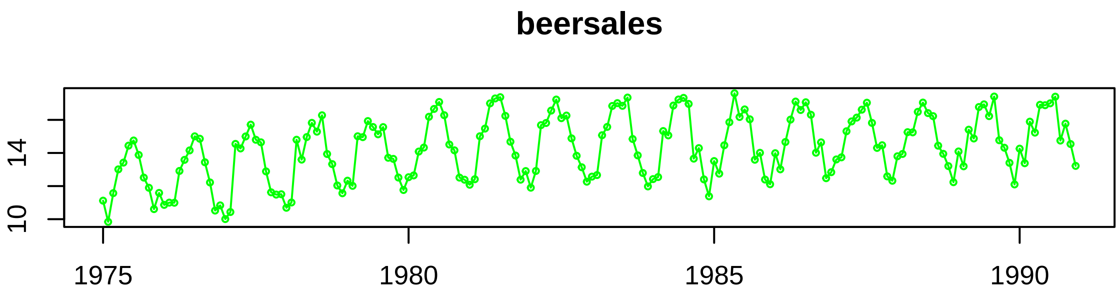
Here we are using the Augmented Dickey-Fuller test for testing the null hypothesis that (alpha, beta, omega) = (aplha, 0, 0) and the alternative hypothesis that (alpha, beta, omega) is not (alpha, 0, 0). We are letting alpha to be estimated freely.

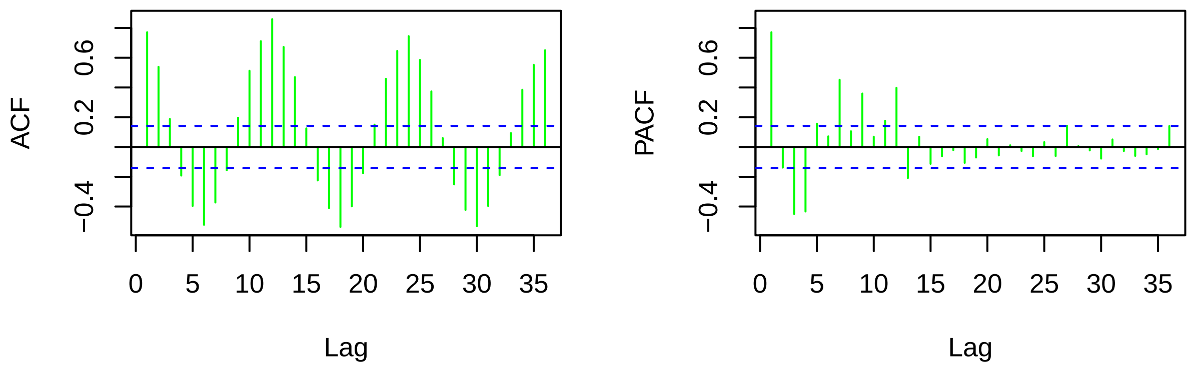
We calculated the value of phi3 as 4.862 in part b and here we observe that critical value for phi 3 is 6.49. So our observed value (of phi 3) 4.862 $<$ the critical value (of phi 3) 6.49 at 5% significance level. Therefore we fail to reject the null hypothesis that (alpha, beta, omega) = (alpha, 0, 0). Which also verifies our claim in part a) that there exist a unit root.

2. Load Beer Sales data (already available on TSA package)

a. Tsdisplay

```r
data(beersales)
tsdisplay(beersales, col = "green")
```






```r
adf.test(beersales)
```

```
## Warning in adf.test(beersales): p-value smaller than printed p-value
```

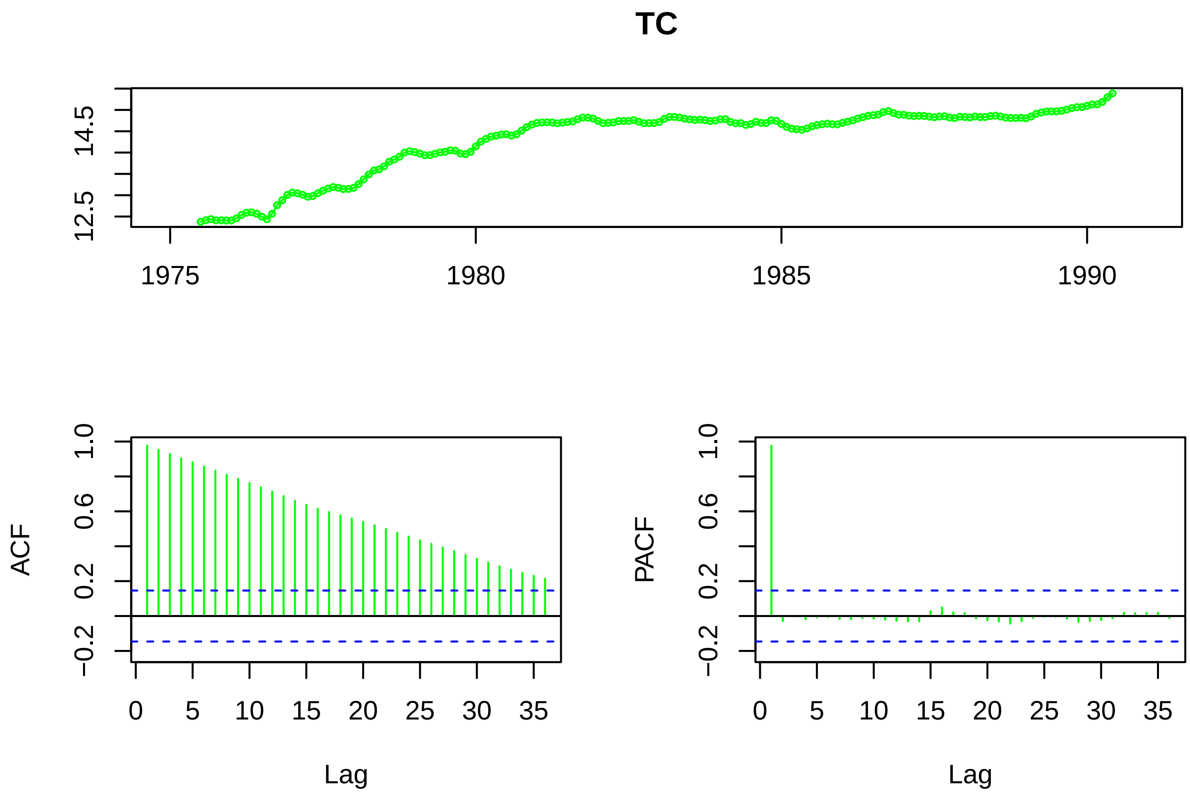
```
##
##  Augmented Dickey-Fuller Test
##
## data:  beersales
## Dickey-Fuller = -9.7734, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

The time series plot of beersales against time clearly shows that there is a seasonal component due to the regularly seen spikes that occur throughout time. There also seem to be a slight upwards trend as well. We can also see the seasonal component by looking at the ACF plot, it kind of follows the graph of a sine function and we know that the sine function is periodic.

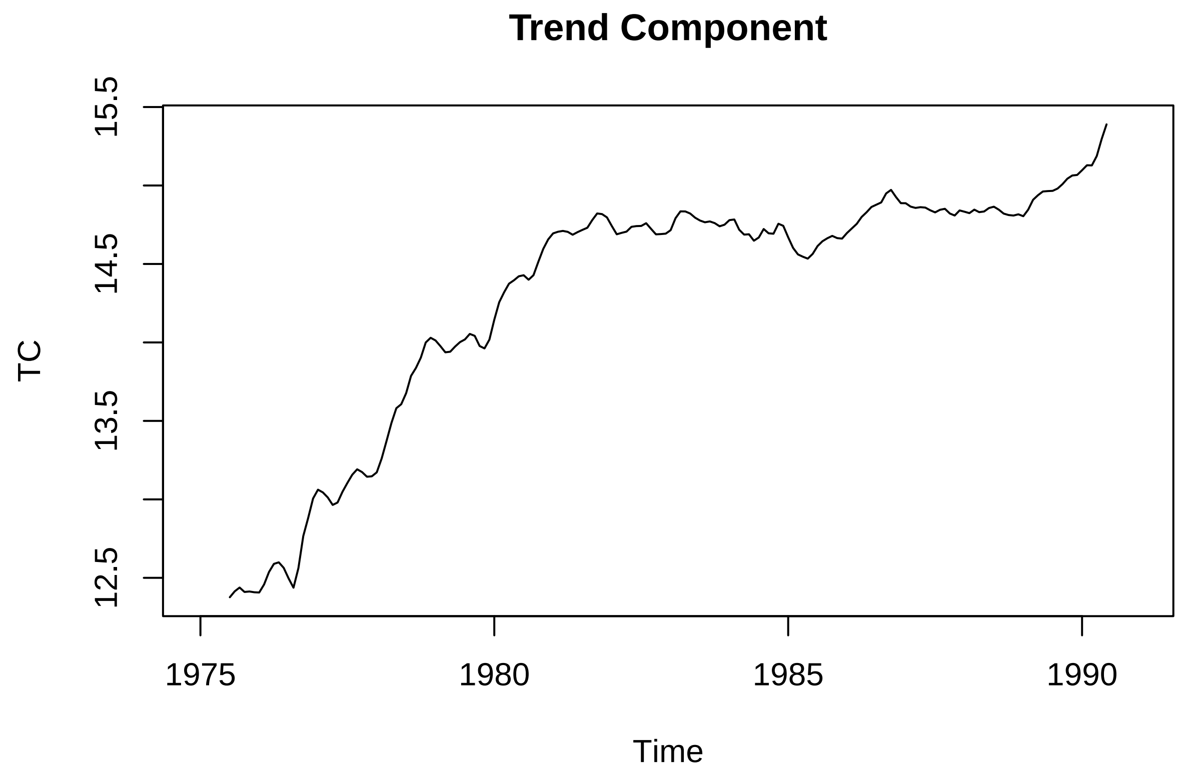
b. Trend estimation and smoothing the series

From the beersales plot in part a, we can see that then spikes occur only once every year. So choosing order 12 for the moving average smoother.

```r
TC=ma(beersales,12)
tsdisplay(TC, col = "green")
```

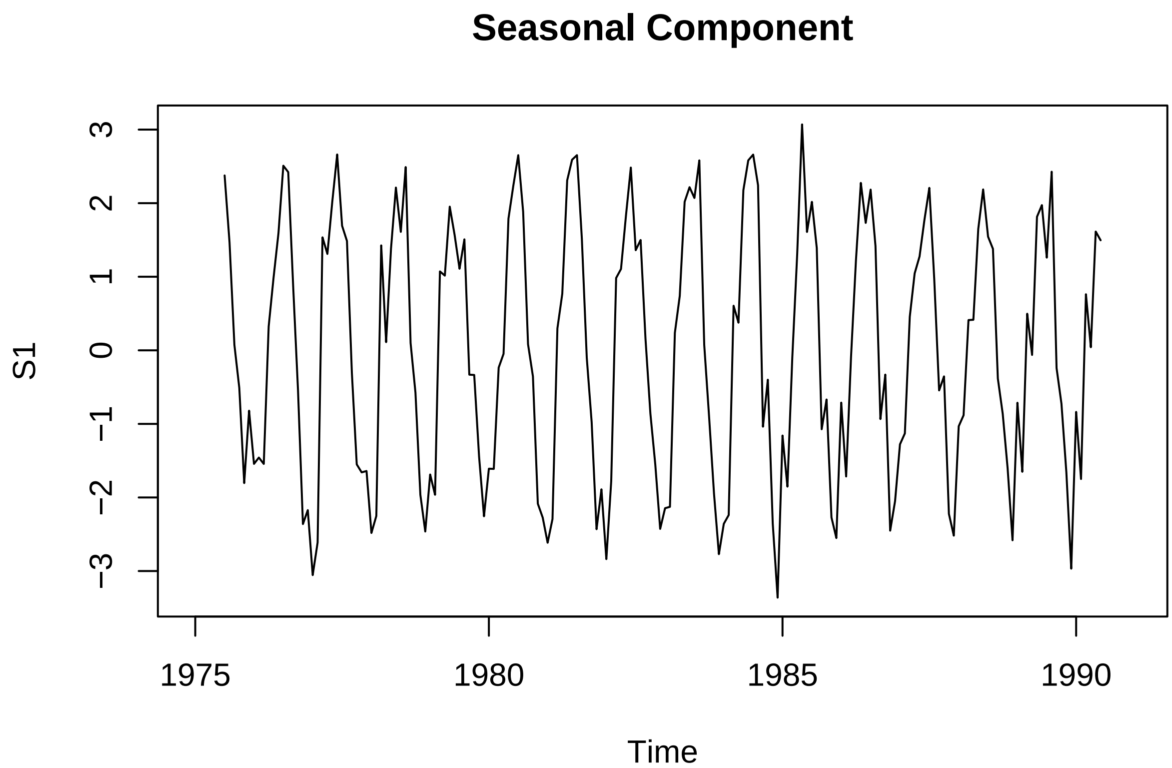



```r
plot(TC, main = "Trend Component")
```

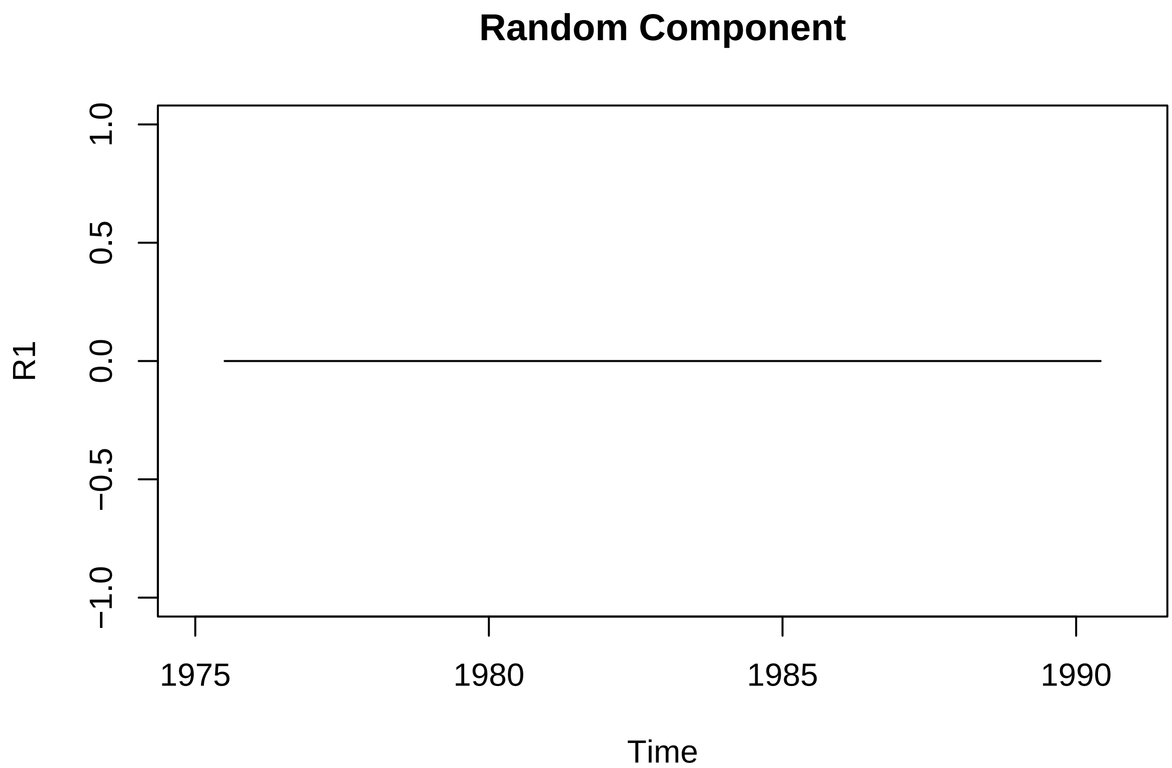


c. Additive Decomposition of the Time Series

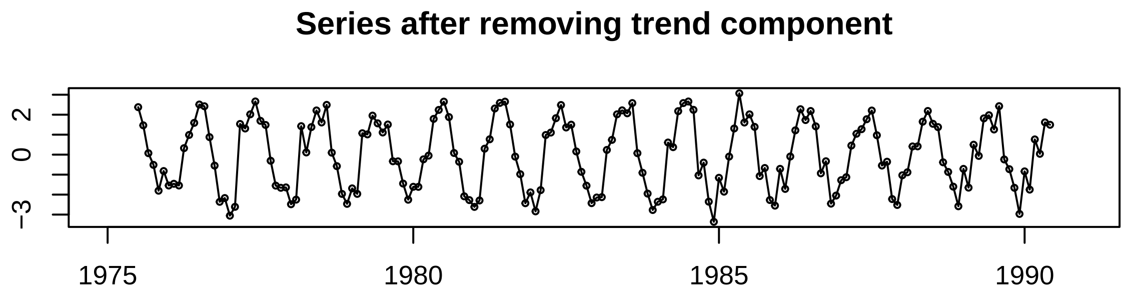
```
# find seasonal and random component and plot them against time
S1 = beersales - TC
R1 = beersales - TC - S1
plot(S1, main = "Seasonal Component")
```

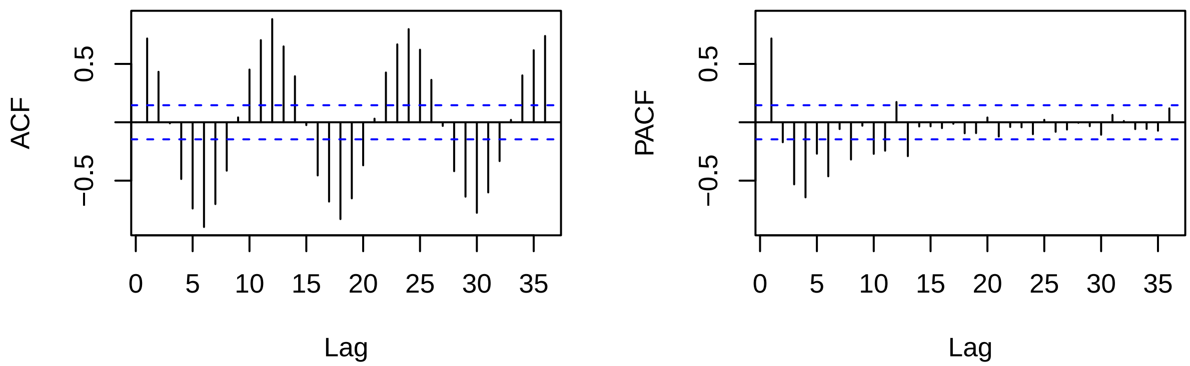



```
plot(R1, main = "Random Component")
```

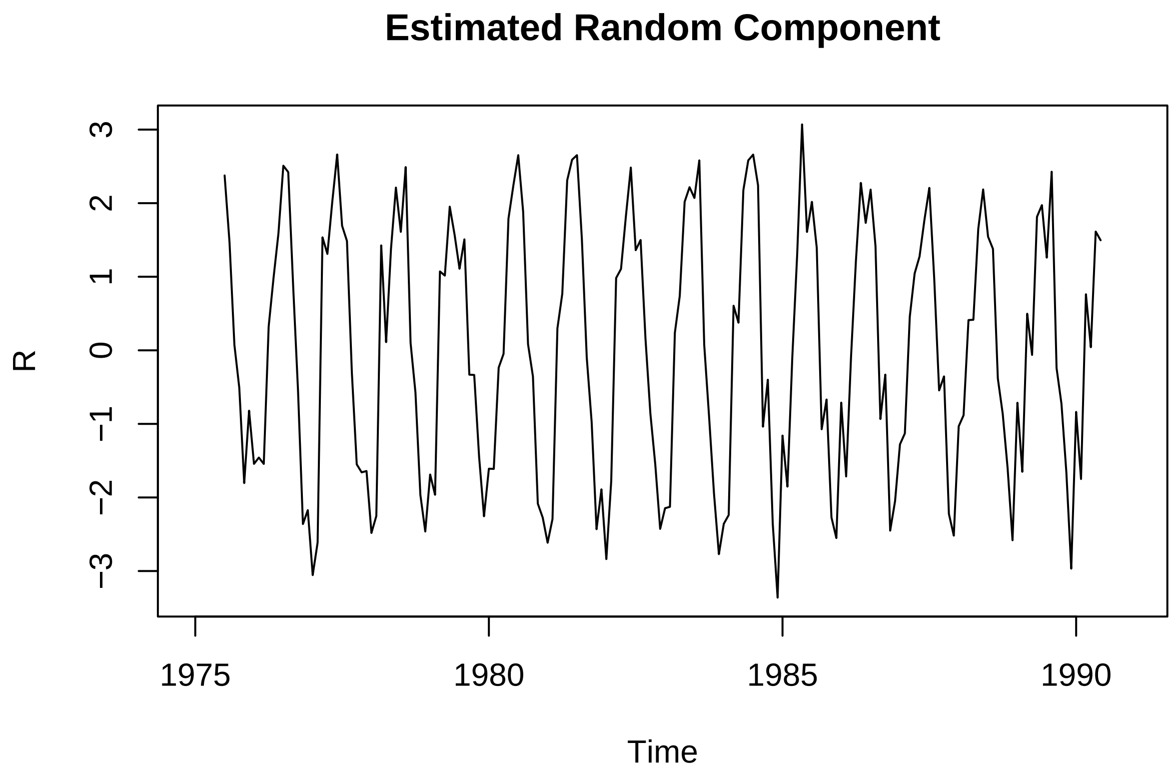



```r
# seasonal component after removing trend component
tsdisplay(beersales-TC ,main = "Series after removing trend component")
```





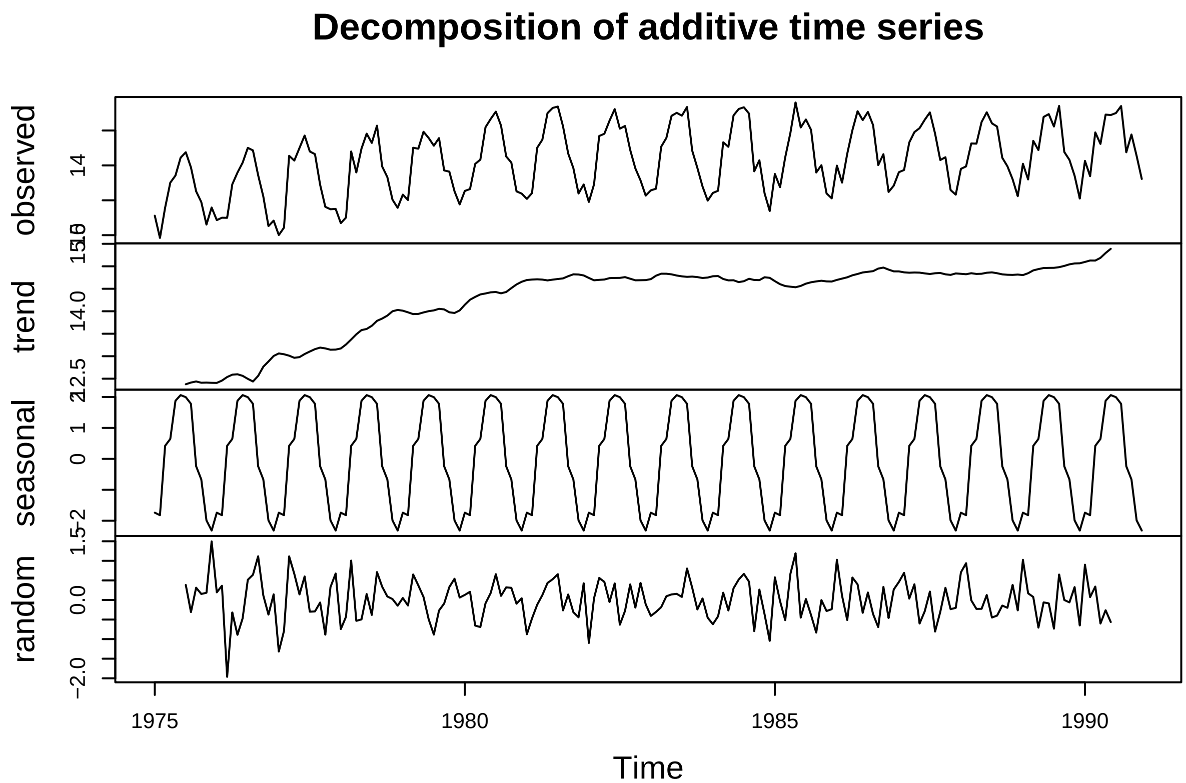

```r
# matrix of seasonal components

pseudo_s=beersales-TC
matrix_s=matrix (pseudo_s, nrow=12)
s=rowMeans (matrix_s, na.rm = TRUE)
srep=rep(length(beersales)/12)
S1 = srep-mean(srep)

# estimate the random component

R=beersales-TC-S1
plot(R, main = "Estimated Random Component")
```




```r
#fitting linear model to trend to forecast
linear_tc=lm (TC~time(beersales))
summary(linear_tc)
```

```
##
## Call:
## lm(formula = TC ~ time(beersales))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.86824 -0.31682 -0.04031  0.34363  0.70449
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -297.29082   13.93976  -21.33   <2e-16 ***
## time(beersales)    0.15714    0.00703   22.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4084 on 178 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.7373, Adjusted R-squared:  0.7359
## F-statistic: 499.7 on 1 and 178 DF,  p-value: < 2.2e-16
```

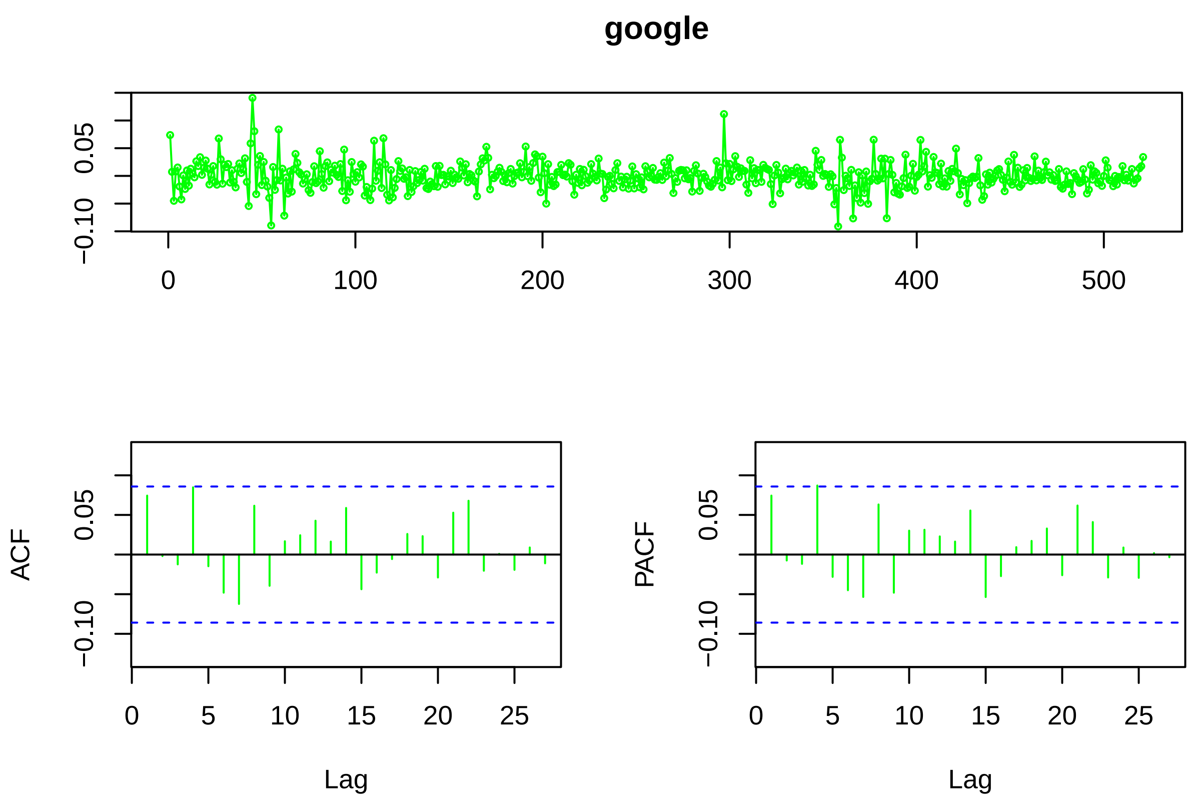
```r
#Verification of the above plots
B=decompose (beersales, type=c("additive"))
plot(B)
```



3. Load Google data (already available on TSA package)

a. Tsdisplay

```r
data(google)
google=google-mean(google)
tsdisplay(google, col = "green")
```




```
adf.test(google)
```

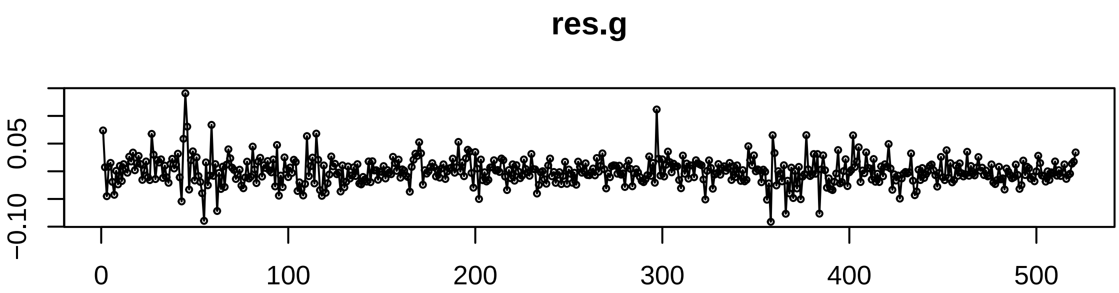
```
## Warning in adf.test(google): p-value smaller than printed p-value
```

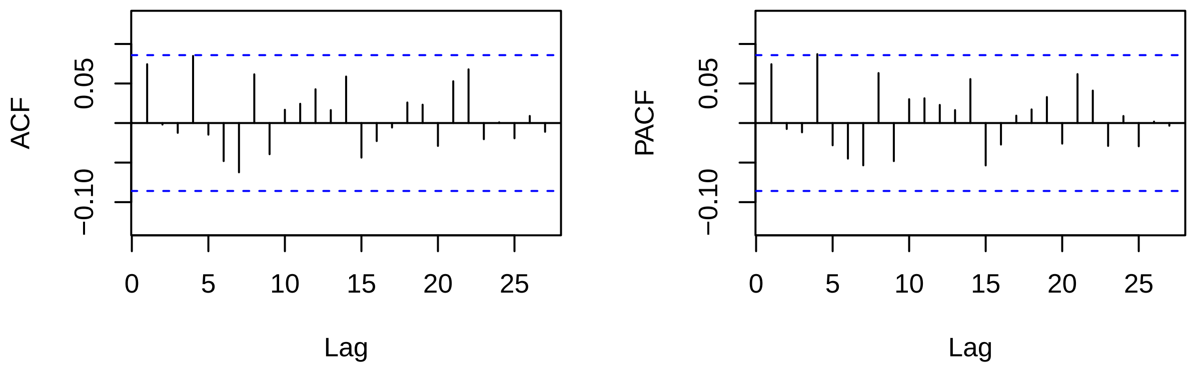
```
##
##  Augmented Dickey-Fuller Test
##
## data:  google
## Dickey-Fuller = -7.982, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

From the above plots of the data, we can see that there are significant spikes which vaguely tells that there are regions of high and low volatility. The ACF and PACF plots show that our data is stationary. By performing an Augmented Dickey-Fuller test in R, we can see that we can reject the null hypothesis that there is a unit root present in our data. Since there is no obvious lags visible poking outside the critical line we take ARMA(0,0) model for the residuals.

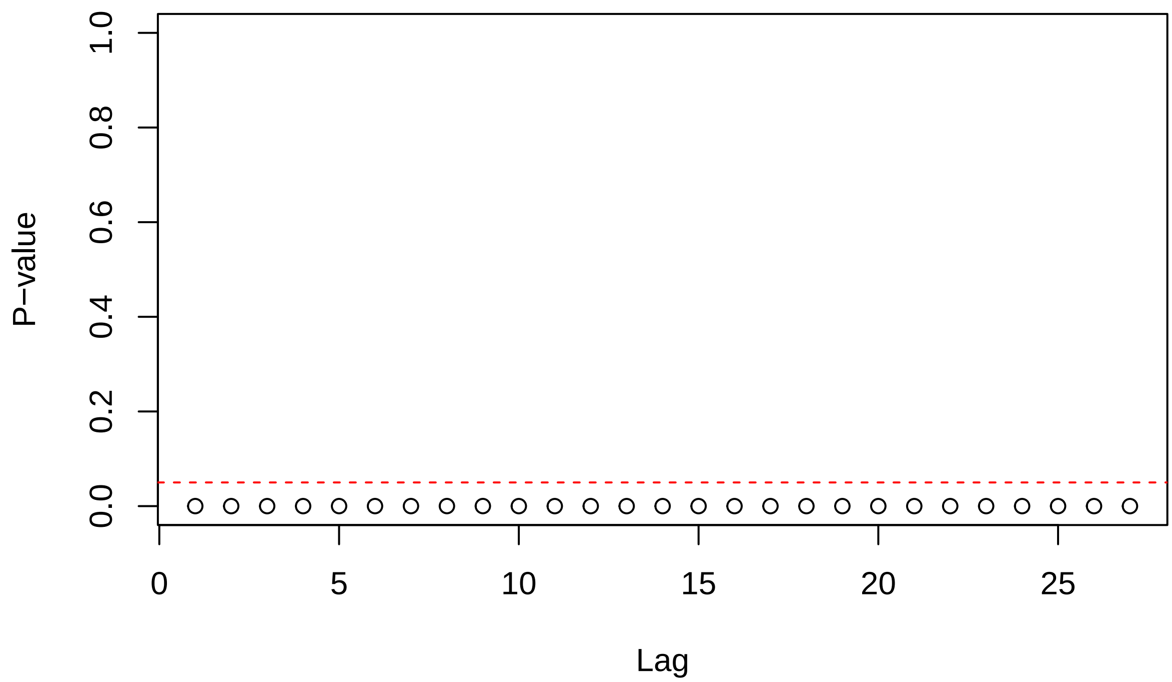
b. Test the data for conditional heteroskedasticity and report the result

```
m.g=arima (google, order=c(0,0,0))
res.g=residuals(m.g)
tsdisplay(res.g)
```

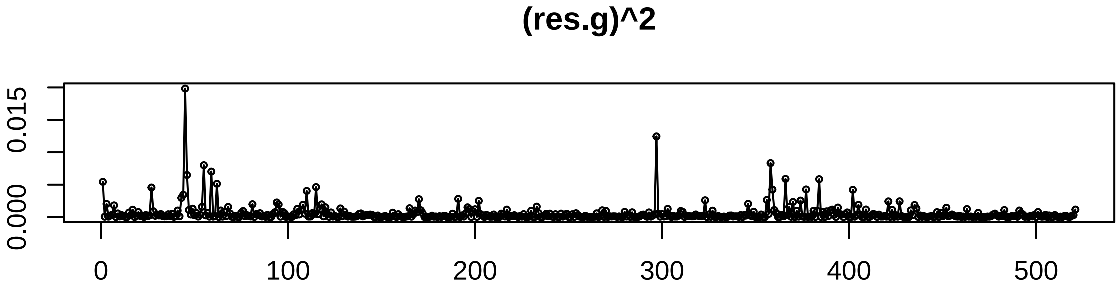


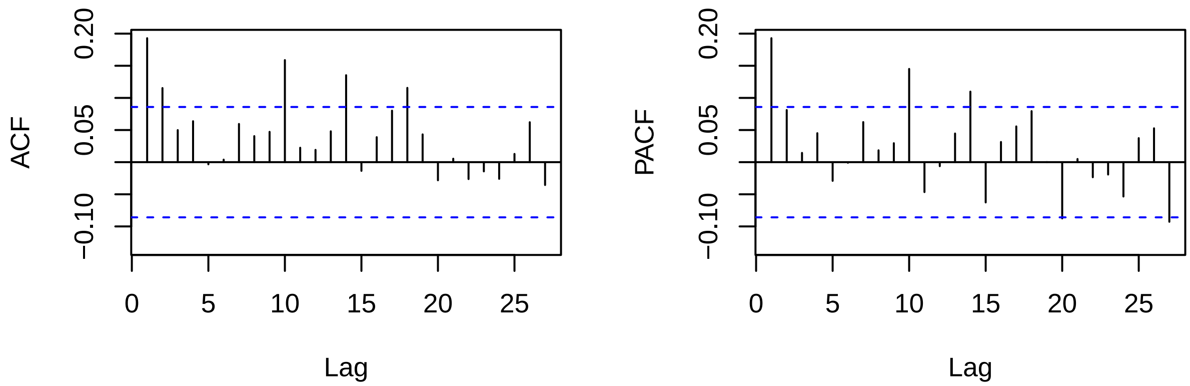



```r
McLeod.Li.test (y=res.g)
```




```r
tsdisplay((res.g)^2)
```





From the ACF and PACF of the residuals we can confirm that there is volatility clustering and our model is a white noise i.e. ARIMA(0,0,0) or ARMA(0,0).

We are using the McLeod - Li test for testing the null hypothesis that: There is Auto Regressive Conditional Heteroscedasticity (ARCH) effect present and the alternative hypothesis that: There is no ARCH effect present.

From the test statistics, We find that all of our points are below the horizontal critical line. Therefore, at 5% significance level it can be said that there is ARCH effect present and we fail to reject the null hypothesis.

The squares of residuals plot shows the high and low volatility regions in the ACF and PACF plots. Knowing that the Squared residuals can be estimated using one of ARMA(1,1) or ARMA(2,1) models also we select these models initially because the ACF and PACF plots of squared residuals shows the significant spikes at various lags which resembles to those models.

c. Determine the ARMA-GARCH model order for the data

```
model1=arima (res.g^2, order=c(1,0,1),include.mean=FALSE)
model2=arima (res.g^2, order=c(2,0,1),include.mean=FALSE)
model3=arima (res.g^2, order=c(1,0,2),include.mean=FALSE)
model4=arima (res.g^2, order=c(2,0,2),include.mean=FALSE)
coeftest(model1)
```

```
##
## z test of coefficients:
##
```

```
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  0.985740   0.013133  75.057 < 2.2e-16 ***
## ma1 -0.914155   0.039651 -23.055 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(model2)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1  1.157350   0.048310  23.9568 < 2.2e-16 ***
## ar2 -0.159751   0.047179  -3.3861  0.000709 ***
## ma1 -0.969473   0.018769 -51.6524 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(model3)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value Pr(>|z|)
## ar1  0.9961225  0.0057143 174.3223  < 2e-16 ***
## ma1 -0.8252196  0.0421506 -19.5779  < 2e-16 ***
## ma2 -0.1325515  0.0428723  -3.0918  0.00199 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(model4)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1  1.43623    0.21148  6.7913 1.112e-11 ***
## ar2 -0.43730    0.21073 -2.0752   0.03797 *
## ma1 -1.25396    0.22296 -5.6242 1.864e-08 ***
## ma2  0.27086    0.21422  1.2644   0.20609
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the coefficient test (z test) it is clear that the choice for selecting ARMA(1,1) and ARMA(2,1) is significant as all the coefficients are statistically different from zero at 5% Significance level.

Now we will fit both of them for our ARMA-GARCH model and try to find the best fit of them by looking at the summary.

```r
fitg=garchFit(~arma(1,1)+garch(1,1),google,include.mean=F)
```

```
##
## Series Initialization:
##  ARMA Model:                arma
##  Formula Mean:              ~ arma(1, 1)
##  GARCH Model:               garch
##  Formula Variance:          ~ garch(1, 1)
##  ARMA Order:                1 1
##  Max ARMA Order:            1
##  GARCH Order:               1 1
##  Max GARCH Order:           1
##  Maximum Order:             1
##  Conditional Dist:          norm
##  h.start:                   2
##  llh.start:                 1
##  Length of Series:          521
##  Recursion Init:            mci
##  Series Scale:              0.02386202
##
## Parameter Initialization:
##  Initial Parameters:          $params
##  Limits of Transformations:   $U, $V
##  Which Parameters are Fixed?  $includes
##  Parameter Matrix:
##                        U            V     params includes
##     mu     -3.033331e-16 3.033331e-16 0.00000000    FALSE
##     ar1    -1.000000e+00 1.000000e+00 0.01371250     TRUE
##     ma1    -1.000000e+00 1.000000e+00 0.06273507     TRUE
##     omega   1.000000e-06 1.000000e+02 0.10000000     TRUE
##     alpha1  1.000000e-08 1.000000e+00 0.10000000     TRUE
##     gamma1 -1.000000e+00 1.000000e+00 0.10000000    FALSE
##     beta1   1.000000e-08 1.000000e+00 0.80000000     TRUE
##     delta   0.000000e+00 2.000000e+00 2.00000000    FALSE
##     skew    1.000000e-01 1.000000e+01 1.00000000    FALSE
##     shape   1.000000e+00 1.000000e+01 4.00000000    FALSE
##  Index List of Parameters to be Optimized:
##    ar1    ma1  omega alpha1  beta1
##      2      3      4      5      7
##  Persistence:                  0.9
##
##
## --- START OF TRACE ---
## Selected Algorithm: nlminb
##
## R coded nlminb Solver:
##
##   0:     705.84940: 0.0137125 0.0627351 0.100000 0.100000 0.800000
##   1:     705.52020: 0.0133736 0.0622999 0.0947242 0.100053 0.796671
##   2:     705.29836: 0.0127583 0.0615040 0.0933998 0.105939 0.798019
##   3:     705.18239: 0.0121967 0.0607626 0.0880758 0.107730 0.795410
##   4:     705.01645: 0.0113893 0.0596285 0.0884898 0.113391 0.797661
##   5:     704.90955: 0.0102188 0.0578086 0.0838634 0.116196 0.795367
```

```
##   6:     704.83821: 0.00903418 0.0550949 0.0844705 0.121681 0.795432
##   7:     704.79450: 0.0105187 0.0538622 0.0838680 0.124686 0.790322
##   8:     704.77553: 0.0123640 0.0515057 0.0864392 0.128016 0.786777
##   9:     704.76669: 0.0136367 0.0455184 0.0854432 0.128586 0.786114
##  10:     704.75730: 0.0193742 0.0448959 0.0843473 0.130660 0.786760
##  11:     704.75503: 0.0225427 0.0417260 0.0858920 0.131794 0.782828
##  12:     704.75085: 0.0247149 0.0371023 0.0885587 0.132991 0.780686
##  13:     704.74549: 0.0279204 0.0322068 0.0869920 0.132188 0.782059
##  14:     704.74283: 0.0327513 0.0289206 0.0859391 0.132034 0.784048
##  15:     704.74111: 0.0375504 0.0253974 0.0865299 0.132406 0.782234
##  16:     704.73899: 0.0413286 0.0206044 0.0874973 0.132966 0.781381
##  17:     704.73824: 0.0424179 0.0179528 0.0868764 0.132948 0.781624
##  18:     704.73692: 0.0448723 0.0168946 0.0866387 0.134075 0.781185
##  19:     704.73670: 0.0469440 0.0148308 0.0867793 0.133982 0.781473
##  20:     704.73641: 0.0491205 0.0128745 0.0868765 0.133865 0.781198
##  21:     704.73628: 0.0497681 0.0115965 0.0867040 0.133997 0.781268
##  22:     704.73624: 0.0507616 0.0105900 0.0868844 0.134127 0.781033
##  23:     704.73624: 0.0510948 0.0102970 0.0868031 0.134040 0.781193
##  24:     704.73624: 0.0514199 0.0100346 0.0868038 0.134115 0.781113
##  25:     704.73624: 0.0513625 0.0100802 0.0868053 0.134101 0.781135
##  26:     704.73624: 0.0513600 0.0100810 0.0868055 0.134099 0.781136
##
## Final Estimate of the Negative LLH:
##  LLH:  -1241.442    norm LLH:  -2.382807
##          ar1          ma1        omega       alpha1        beta1
## 5.136001e-02 1.008103e-02 4.942666e-05 1.340994e-01 7.811355e-01
##
## R-optimhess Difference Approximated Hessian Matrix:
##                 ar1           ma1         omega        alpha1         beta1
## ar1      -417.046015   -401.650705 -2.029852e+04 -1.310176e+01 -9.818961e+00
## ma1      -401.650705   -412.401090 -3.197780e+04 -9.575956e+00 -1.307483e+01
## omega  -20298.517339 -31977.802587 -2.898936e+10 -8.500497e+06 -1.275028e+07
## alpha1    -13.101759     -9.575956 -8.500497e+06 -4.109819e+03 -4.626132e+03
## beta1      -9.818961    -13.074827 -1.275028e+07 -4.626132e+03 -6.352704e+03
## attr(,"time")
## Time difference of 0.02332592 secs
##
## --- END OF TRACE ---
##
##
## Time to Estimate Parameters:
##  Time difference of 0.259089 secs
```

```r
summary(fitg)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 1) + garch(1, 1), data = google,
##     include.mean = F)
##
## Mean and Variance Equation:
```

```
##  data ~ arma(1, 1) + garch(1, 1)
## <environment: 0x000000fd83962fd8>
##  [data = google]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         ar1          ma1        omega       alpha1        beta1
## 5.1360e-02  1.0081e-02  4.9427e-05  1.3410e-01  7.8114e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## ar1    5.136e-02   1.968e-01    0.261  0.79412
## ma1    1.008e-02   1.979e-01    0.051  0.95937
## omega  4.943e-05   1.973e-05    2.505  0.01225 *
## alpha1 1.341e-01   4.229e-02    3.171  0.00152 **
## beta1  7.811e-01   6.229e-02   12.540  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1241.442    normalized:  2.382807
##
## Description:
##  Thu Jul 20 13:22:11 2023 by user: Anisha
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  197.341   0
##  Shapiro-Wilk Test  R    W      0.969496  6.107007e-09
##  Ljung-Box Test     R    Q(10)  11.60095  0.3126504
##  Ljung-Box Test     R    Q(15)  15.67283  0.4041226
##  Ljung-Box Test     R    Q(20)  16.35514  0.6943622
##  Ljung-Box Test     R^2  Q(10)  5.265711  0.8727361
##  Ljung-Box Test     R^2  Q(15)  8.598789  0.8975444
##  Ljung-Box Test     R^2  Q(20)  13.23465  0.8670906
##  LM Arch Test       R    TR^2   5.251862  0.9490331
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -4.746420 -4.705577 -4.746601 -4.730421
```

Here, the summary shows that the ARMA parameters are statistically significant and not different from zero while the GARCH parameters are significantly different than zero.

```
fitg1=garchFit (~arma (2,1) +garch(1,2),google, include.mean=F)
```

```
##
## Series Initialization:
##  ARMA Model:                arma
##  Formula Mean:              ~ arma(2, 1)
##  GARCH Model:               garch
##  Formula Variance:          ~ garch(1, 2)
##  ARMA Order:                2 1
##  Max ARMA Order:            2
##  GARCH Order:               1 2
##  Max GARCH Order:           2
##  Maximum Order:             2
##  Conditional Dist:          norm
##  h.start:                   3
##  llh.start:                 1
##  Length of Series:          521
##  Recursion Init:            mci
##  Series Scale:              0.02386202
##
## Parameter Initialization:
##  Initial Parameters:          $params
##  Limits of Transformations:   $U, $V
##  Which Parameters are Fixed?  $includes
##  Parameter Matrix:
##                        U            V     params includes
##     mu     -3.033331e-16 3.033331e-16  0.0000000    FALSE
##     ar1    -1.000000e+00 1.000000e+00 -0.8105180     TRUE
##     ar2    -1.000000e+00 1.000000e+00  0.1012434     TRUE
##     ma1    -1.000000e+00 1.000000e+00  0.8893363     TRUE
##     omega   1.000000e-06 1.000000e+02  0.1000000     TRUE
##     alpha1  1.000000e-08 1.000000e+00  0.1000000     TRUE
##     gamma1 -1.000000e+00 1.000000e+00  0.1000000    FALSE
##     beta1   1.000000e-08 1.000000e+00  0.4000000     TRUE
##     beta2   1.000000e-08 1.000000e+00  0.4000000     TRUE
##     delta   0.000000e+00 2.000000e+00  2.0000000    FALSE
##     skew    1.000000e-01 1.000000e+01  1.0000000    FALSE
##     shape   1.000000e+00 1.000000e+01  4.0000000    FALSE
##  Index List of Parameters to be Optimized:
##    ar1    ar2    ma1  omega alpha1  beta1  beta2
##      2      3      4      5      6      8      9
##  Persistence:                  0.9
##
##
## --- START OF TRACE ---
## Selected Algorithm: nlminb
##
## R coded nlminb Solver:
##
##   0:     705.87464: -0.810518 0.101243 0.889336 0.100000 0.100000 0.400000 0.400000
##   1:     705.41385: -0.810970 0.101049 0.888732 0.0961068 0.0993473 0.396959 0.396714
##   2:     704.97950: -0.814016 0.0988748 0.884283 0.0922681 0.114268 0.397647 0.395411
##   3:     704.94326: -0.811115 0.0897724 0.884589 0.0850301 0.120336 0.392487 0.387278
##   4:     704.52029: -0.812522 0.0876668 0.881956 0.0903361 0.124960 0.394787 0.387678
##   5:     704.43755: -0.816977 0.0889069 0.876756 0.0905403 0.124472 0.392882 0.383676
##   6:     704.30506: -0.816956 0.0839952 0.874948 0.0943179 0.128737 0.393881 0.380942
```

```
##   7:     704.21956: -0.816633 0.0792693 0.873684 0.0944365 0.131616 0.392169 0.375192
##   8:     704.12386: -0.817947 0.0773231 0.871372 0.0979747 0.135431 0.394351 0.370128
##   9:     704.06285: -0.819304 0.0777499 0.869666 0.0982302 0.137100 0.396270 0.362586
##  10:     703.99220: -0.818699 0.0766114 0.870008 0.0994484 0.140586 0.400835 0.356922
##  11:     703.94831: -0.819247 0.0746976 0.868812 0.0991254 0.141794 0.404446 0.349969
##  12:     703.90050: -0.820171 0.0739942 0.867503 0.101074 0.142056 0.410135 0.344580
##  13:     703.63101: -0.814602 0.0928319 0.879609 0.103673 0.149818 0.489241 0.255397
##  14:     703.03782: -0.819527 0.0674594 0.862274 0.0960915 0.139582 0.669299 0.0921531
##  15:     703.02744: -0.820462 0.0683766 0.861434 0.0975412 0.140916 0.670732 0.0934546
##  16:     702.99887: -0.821060 0.0685824 0.860910 0.0955310 0.140171 0.671854 0.0915786
##  17:     702.99535: -0.821715 0.0685834 0.860774 0.0945739 0.140337 0.674700 0.0908309
##  18:     702.99349: -0.823096 0.0683034 0.861464 0.0941070 0.139946 0.676624 0.0889568
##  19:     702.99082: -0.824509 0.0680547 0.862740 0.0958764 0.140004 0.676868 0.0871770
##  20:     702.98832: -0.824174 0.0678376 0.863936 0.0947138 0.141053 0.678541 0.0853859
##  21:     702.98793: -0.823341 0.0684306 0.862980 0.0947705 0.141364 0.681151 0.0843014
##  22:     702.98309: -0.822652 0.0687486 0.862233 0.0943810 0.141035 0.681517 0.0833089
##  23:     702.98198: -0.822094 0.0690401 0.861684 0.0944645 0.141004 0.682603 0.0825172
##  24:     702.97063: -0.818088 0.0709565 0.860550 0.0900783 0.136470 0.710038 0.0637820
##  25:     702.96509: -0.819443 0.0723840 0.857877 0.0906507 0.135499 0.733165 0.0389366
##  26:     702.93023: -0.831705 0.0686271 0.869027 0.0922293 0.138653 0.752977 0.0172398
##  27:     702.92526: -0.822580 0.0696439 0.863993 0.0943997 0.139764 0.755880 0.0116243
##  28:     702.92020: -0.820208 0.0709975 0.861143 0.0942089 0.139120 0.764214 0.00325458
##  29:     702.91815: -0.823826 0.0703562 0.863956 0.0922886 0.134695 0.773578 1.00000e-08
##  30:     702.91716: -0.823656 0.0699133 0.863422 0.0909979 0.135143 0.774217 1.00000e-08
##  31:     702.91713: -0.822602 0.0700675 0.862334 0.0909700 0.135735 0.774532 1.00000e-08
##  32:     702.91641: -0.822505 0.0702968 0.862560 0.0912189 0.135886 0.773829 1.00000e-08
##  33:     702.91624: -0.822772 0.0703849 0.863017 0.0915353 0.136098 0.773318 1.00000e-08
##  34:     702.91616: -0.822388 0.0706262 0.862806 0.0919000 0.136907 0.772144 1.00000e-08
##  35:     702.91613: -0.823293 0.0702979 0.863551 0.0917965 0.136661 0.772486 1.00000e-08
##  36:     702.91613: -0.822688 0.0705448 0.863070 0.0918585 0.136533 0.772565 1.00000e-08
##  37:     702.91611: -0.822848 0.0704714 0.863191 0.0918331 0.136641 0.772477 1.00000e-08
##  38:     702.91611: -0.822867 0.0704633 0.863207 0.0918343 0.136634 0.772482 1.00000e-08
##
## Final Estimate of the Negative LLH:
##  LLH:  -1243.262    norm LLH:  -2.3863
##           ar1           ar2           ma1         omega        alpha1
## -8.228674e-01  7.046328e-02  8.632066e-01  5.229004e-05  1.366345e-01
##         beta1         beta2
##  7.724823e-01  1.000000e-08
##
## R-optimhess Difference Approximated Hessian Matrix:
##                  ar1         ar2          ma1         omega        alpha1
## ar1      -1955.83751  1723.95068  -1750.66332 -1.062807e+05  4.839550e+01
## ar2       1723.95068 -1988.26804   1492.73287  9.952834e+04 -9.031280e+01
## ma1      -1750.66332  1492.73287  -1718.80688 -5.255050e+04  4.146689e+01
## omega  -106280.68276 99528.33786 -52550.49997 -2.708019e+10 -7.856833e+06
## alpha1      48.39550   -90.31280     41.46689 -7.856833e+06 -3.810706e+03
## beta1       35.86549   -49.58189     48.30381 -1.186664e+07 -4.260106e+03
## beta2       51.27902   -71.26212     62.08356 -1.204701e+07 -4.302108e+03
##                beta1         beta2
## ar1     3.586549e+01  5.127902e+01
## ar2    -4.958189e+01 -7.126212e+01
## ma1     4.830381e+01  6.208356e+01
## omega  -1.186664e+07 -1.204701e+07
```

```
## alpha1 -4.260106e+03 -4.302108e+03
## beta1  -5.888680e+03 -5.993988e+03
## beta2  -5.993988e+03 -6.118635e+03
## attr(,"time")
## Time difference of 0.03568912 secs
##
## --- END OF TRACE ---
##
##
## Time to Estimate Parameters:
##  Time difference of 0.1340661 secs
```

```
summary(fitg1)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(2, 1) + garch(1, 2), data = google,
##     include.mean = F)
##
## Mean and Variance Equation:
##  data ~ arma(2, 1) + garch(1, 2)
## <environment: 0x000000fd8129f080>
##  [data = google]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         ar1          ar2          ma1        omega       alpha1        beta1
## -0.82286738   0.07046328   0.86320660   0.00005229   0.13663447   0.77248228
##       beta2
##  0.00000001
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## ar1    -8.229e-01   9.421e-02   -8.735  < 2e-16 ***
## ar2     7.046e-02   4.740e-02    1.486  0.13716
## ma1     8.632e-01   8.271e-02   10.437  < 2e-16 ***
## omega   5.229e-05   2.152e-05    2.430  0.01512 *
## alpha1  1.366e-01   4.680e-02    2.919  0.00351 **
## beta1   7.725e-01   3.092e-01    2.498  0.01249 *
## beta2   1.000e-08   2.690e-01    0.000  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1243.262    normalized:  2.3863
##
```

```
## Description:
##  Thu Jul 20 13:22:11 2023 by user: Anisha
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  200.4352  0
##  Shapiro-Wilk Test  R    W      0.9690541 4.981861e-09
##  Ljung-Box Test     R    Q(10)  8.161138  0.6131006
##  Ljung-Box Test     R    Q(15)  11.9095   0.6858656
##  Ljung-Box Test     R    Q(20)  12.93879  0.8799948
##  Ljung-Box Test     R^2  Q(10)  5.119039  0.8830854
##  Ljung-Box Test     R^2  Q(15)  8.616752  0.896689
##  Ljung-Box Test     R^2  Q(20)  12.97446  0.878477
##  LM Arch Test       R    TR^2   5.052846  0.9561905
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -4.745729 -4.688550 -4.746084 -4.723332
```

Here except for the AR(2) parameter, all other parameters are statistically significant.

Earlier choice of ARMA(0,0) by looking at the stationary series and now we find the p and q for GARCH from the models above is significant in both case regardless of the ARMA(1,1) or ARMA(2,1) we try to fit them with ARMA(0,0).

So the models we have results in

ARMA(0,0) gives GARCH(1,1)

ARMA(0,0) give GARCH(1,2)

```
fitg2=garchFit(~arma(0,0) +garch(1,1),google,include.mean=F)
```

```
##
## Series Initialization:
##  ARMA Model:                arma
##  Formula Mean:              ~ arma(0, 0)
##  GARCH Model:               garch
##  Formula Variance:          ~ garch(1, 1)
##  ARMA Order:                0 0
##  Max ARMA Order:            0
##  GARCH Order:               1 1
##  Max GARCH Order:           1
##  Maximum Order:             1
##  Conditional Dist:          norm
##  h.start:                   2
##  llh.start:                 1
##  Length of Series:          521
##  Recursion Init:            mci
##  Series Scale:              0.02386202
##
```

```
## Parameter Initialization:
##  Initial Parameters:          $params
##  Limits of Transformations:   $U, $V
##  Which Parameters are Fixed?  $includes
##  Parameter Matrix:
##                      U           V params includes
##     mu     -3.033331e-16 3.033331e-16    0.0    FALSE
##     omega   1.000000e-06 1.000000e+02    0.1     TRUE
##     alpha1  1.000000e-08 1.000000e+00    0.1     TRUE
##     gamma1 -1.000000e+00 1.000000e+00    0.1    FALSE
##     beta1   1.000000e-08 1.000000e+00    0.8     TRUE
##     delta   0.000000e+00 2.000000e+00    2.0    FALSE
##     skew    1.000000e-01 1.000000e+01    1.0    FALSE
##     shape   1.000000e+00 1.000000e+01    4.0    FALSE
##  Index List of Parameters to be Optimized:
##  omega alpha1  beta1
##      2      3      5
##  Persistence:                  0.9
##
##
## --- START OF TRACE ---
## Selected Algorithm: nlminb
##
## R coded nlminb Solver:
##
##   0:     711.39951: 0.100000 0.100000 0.800000
##   1:     711.06655: 0.0947083 0.100134 0.796505
##   2:     710.84400: 0.0935107 0.106249 0.797692
##   3:     710.73101: 0.0881966 0.108077 0.794751
##   4:     710.55869: 0.0887568 0.114135 0.796545
##   5:     710.45466: 0.0845216 0.117557 0.793291
##   6:     710.37971: 0.0862378 0.123435 0.791635
##   7:     710.33377: 0.0866964 0.126071 0.785884
##   8:     710.30913: 0.0896680 0.129842 0.781739
##   9:     710.28271: 0.0883786 0.134393 0.777512
##  10:     710.27401: 0.0924091 0.136417 0.773052
##  11:     710.26990: 0.0904816 0.137045 0.774409
##  12:     710.26856: 0.0901808 0.138756 0.772697
##  13:     710.26721: 0.0904553 0.139900 0.772806
##  14:     710.26554: 0.0910220 0.139775 0.771777
##  15:     710.26499: 0.0917082 0.139912 0.770825
##  16:     710.26494: 0.0917180 0.141003 0.770374
##  17:     710.26460: 0.0915917 0.140896 0.770218
##  18:     710.26418: 0.0925103 0.141219 0.768934
##  19:     710.26413: 0.0926049 0.141702 0.768507
##  20:     710.26412: 0.0926970 0.141712 0.768359
##  21:     710.26412: 0.0926983 0.141706 0.768372
##  22:     710.26412: 0.0926965 0.141707 0.768373
##
## Final Estimate of the Negative LLH:
##  LLH:  -1235.914    norm LLH:  -2.372197
##        omega       alpha1        beta1
## 5.278098e-05 1.417067e-01 7.683728e-01
##
```

```
## R-optimhess Difference Approximated Hessian Matrix:
##               omega       alpha1         beta1
## omega  -25787493490 -7496725.089 -11412321.003
## alpha1     -7496725    -3657.487     -4145.603
## beta1     -11412321    -4145.603     -5756.699
## attr(,"time")
## Time difference of 0.005745173 secs
##
## --- END OF TRACE ---
##
##
## Time to Estimate Parameters:
##  Time difference of 0.03030014 secs
```

```r
summary(fitg2)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 0) + garch(1, 1), data = google,
##     include.mean = F)
##
## Mean and Variance Equation:
##  data ~ arma(0, 0) + garch(1, 1)
## <environment: 0x000000fd8b688a10>
##  [data = google]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##      omega      alpha1       beta1
## 5.2781e-05  1.4171e-01  7.6837e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  5.278e-05   2.046e-05    2.579  0.00990 **
## alpha1 1.417e-01   4.439e-02    3.192  0.00141 **
## beta1  7.684e-01   6.423e-02   11.964  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1235.914    normalized:  2.372197
##
## Description:
##  Thu Jul 20 13:22:11 2023 by user: Anisha
##
##
```

```
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  197.9118  0
##  Shapiro-Wilk Test  R    W      0.9683952 3.688882e-09
##  Ljung-Box Test     R    Q(10)  11.78695  0.2995681
##  Ljung-Box Test     R    Q(15)  16.42109  0.3546311
##  Ljung-Box Test     R    Q(20)  17.22032  0.6386243
##  Ljung-Box Test     R^2  Q(10)  5.091376  0.8849909
##  Ljung-Box Test     R^2  Q(15)  8.717806  0.8918042
##  Ljung-Box Test     R^2  Q(20)  13.24386  0.8666776
##  LM Arch Test       R    TR^2   5.023359  0.9571942
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -4.732877 -4.708372 -4.732943 -4.723278
```

Here, the summary shows that all the parameters of ARMA and GARCH are statistically significant.

```
fitg3=garchFit(~arma(0,0) +garch(1,2),google,include.mean=F)
```

```
##
## Series Initialization:
##  ARMA Model:                arma
##  Formula Mean:              ~ arma(0, 0)
##  GARCH Model:               garch
##  Formula Variance:          ~ garch(1, 2)
##  ARMA Order:                0 0
##  Max ARMA Order:            0
##  GARCH Order:               1 2
##  Max GARCH Order:           2
##  Maximum Order:             2
##  Conditional Dist:          norm
##  h.start:                   3
##  llh.start:                 1
##  Length of Series:          521
##  Recursion Init:            mci
##  Series Scale:              0.02386202
##
## Parameter Initialization:
##  Initial Parameters:          $params
##  Limits of Transformations:   $U, $V
##  Which Parameters are Fixed?  $includes
##  Parameter Matrix:
##                      U            V params includes
##     mu     -3.033331e-16 3.033331e-16    0.0    FALSE
##     omega   1.000000e-06 1.000000e+02    0.1     TRUE
##     alpha1  1.000000e-08 1.000000e+00    0.1     TRUE
##     gamma1 -1.000000e+00 1.000000e+00    0.1    FALSE
##     beta1   1.000000e-08 1.000000e+00    0.4     TRUE
##     beta2   1.000000e-08 1.000000e+00    0.4     TRUE
##     delta   0.000000e+00 2.000000e+00    2.0    FALSE
```

```
##     skew    1.000000e-01 1.000000e+01    1.0    FALSE
##     shape   1.000000e+00 1.000000e+01    4.0    FALSE
##  Index List of Parameters to be Optimized:
##  omega alpha1  beta1  beta2
##      2      3      5      6
##  Persistence:                  0.9
##
##
## --- START OF TRACE ---
## Selected Algorithm: nlminb
##
## R coded nlminb Solver:
##
##   0:     712.79953: 0.100000 0.100000 0.400000 0.400000
##   1:     712.41717: 0.0960963 0.0999178 0.397109 0.396824
##   2:     712.18192: 0.0951223 0.105597 0.397813 0.396870
##   3:     712.05150: 0.0912483 0.107814 0.395528 0.393946
##   4:     711.84611: 0.0920796 0.113333 0.397055 0.394401
##   5:     711.71024: 0.0892705 0.116427 0.395085 0.390886
##   6:     711.57333: 0.0909259 0.121722 0.396086 0.389502
##   7:     711.46951: 0.0903836 0.124573 0.394447 0.384750
##   8:     711.38278: 0.0929815 0.128727 0.395376 0.381778
##   9:     711.31016: 0.0932377 0.130929 0.394841 0.376440
##  10:     711.24573: 0.0953375 0.134171 0.397145 0.372770
##  11:     711.19003: 0.0949076 0.135684 0.398579 0.367370
##  12:     711.14130: 0.0962079 0.138006 0.402104 0.363603
##  13:     711.09849: 0.0958895 0.138949 0.404262 0.358307
##  14:     711.05953: 0.0971382 0.140703 0.408000 0.354423
##  15:     711.02376: 0.0966407 0.141319 0.410704 0.349348
##  16:     710.99040: 0.0975160 0.142696 0.414804 0.345578
##  17:     710.95922: 0.0971285 0.143093 0.417706 0.340581
##  18:     710.92972: 0.0980303 0.144222 0.421824 0.336754
##  19:     710.90182: 0.0974581 0.144457 0.425031 0.331955
##  20:     710.87520: 0.0981294 0.145377 0.429331 0.328225
##  21:     710.84978: 0.0977950 0.145497 0.432533 0.323397
##  22:     710.82543: 0.0984229 0.146270 0.436869 0.319668
##  23:     710.80216: 0.0977115 0.146273 0.440413 0.315126
##  24:     710.77962: 0.0984782 0.146914 0.444667 0.311306
##  25:     710.75796: 0.0981753 0.146862 0.448050 0.306599
##  26:     710.73736: 0.0980584 0.147385 0.452804 0.303311
##  27:     710.71702: 0.0979206 0.147272 0.456158 0.298577
##  28:     710.69781: 0.0993247 0.147774 0.459956 0.294448
##  29:     710.67959: 0.0970222 0.147700 0.464281 0.291336
##  30:     710.66205: 0.0976493 0.148325 0.468673 0.287646
##  31:     710.64316: 0.0994913 0.148249 0.470289 0.282384
##  32:     710.62470: 0.0991059 0.149480 0.474943 0.279163
##  33:     710.60774: 0.0967627 0.148924 0.479156 0.275978
##  34:     710.53182: 0.100620 0.144249 0.509886 0.245047
##  35:     710.52553: 0.0956463 0.157797 0.512679 0.242981
##  36:     710.46293: 0.0933600 0.155514 0.516512 0.237508
##  37:     710.44707: 0.0931729 0.154473 0.522819 0.233741
##  38:     710.43419: 0.0931270 0.152878 0.527869 0.228541
##  39:     710.42049: 0.0928610 0.152067 0.534148 0.224676
##  40:     710.36697: 0.0879640 0.145594 0.574110 0.194189
```

```
##  41:     710.25102: 0.0954540 0.151288 0.602742 0.153153
##  42:     710.17627: 0.0935541 0.148210 0.678568 0.0852797
##  43:     710.17338: 0.0919714 0.145061 0.743485 0.0221339
##  44:     710.15728: 0.0915868 0.143959 0.724840 0.0436388
##  45:     710.15721: 0.0912419 0.144091 0.726483 0.0422781
##  46:     710.15720: 0.0912547 0.143964 0.726905 0.0419288
##  47:     710.15720: 0.0912572 0.143973 0.726852 0.0419745
##
## Final Estimate of the Negative LLH:
##  LLH:  -1236.021    norm LLH:  -2.372402
##        omega       alpha1        beta1        beta2
## 5.196147e-05 1.439731e-01 7.268515e-01 4.197445e-02
##
## R-optimhess Difference Approximated Hessian Matrix:
##               omega       alpha1         beta1         beta2
## omega  -25758791474 -7467950.499 -11388812.882 -11567900.875
## alpha1     -7467950    -3586.779     -4110.108     -4150.457
## beta1     -11388813    -4110.108     -5735.821     -5839.907
## beta2     -11567901    -4150.457     -5839.907     -5963.842
## attr(,"time")
## Time difference of 0.01224995 secs
##
## --- END OF TRACE ---
##
##
## Time to Estimate Parameters:
##  Time difference of 0.07359195 secs
```

```r
summary(fitg3)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 0) + garch(1, 2), data = google,
##     include.mean = F)
##
## Mean and Variance Equation:
##  data ~ arma(0, 0) + garch(1, 2)
## <environment: 0x000000fd8c5e0a68>
##  [data = google]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##      omega      alpha1       beta1       beta2
## 5.1961e-05  1.4397e-01  7.2685e-01  4.1974e-02
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
```

```
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  5.196e-05   2.162e-05    2.403  0.01625 *
## alpha1 1.440e-01   5.017e-02    2.870  0.00411 **
## beta1  7.269e-01   3.049e-01    2.384  0.01713 *
## beta2  4.197e-02   2.636e-01    0.159  0.87347
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1236.021    normalized:  2.372402
##
## Description:
##  Thu Jul 20 13:22:11 2023 by user: Anisha
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  193.9888  0
##  Shapiro-Wilk Test  R    W      0.9689277 4.701412e-09
##  Ljung-Box Test     R    Q(10)  11.95814  0.287868
##  Ljung-Box Test     R    Q(15)  16.30686  0.3619562
##  Ljung-Box Test     R    Q(20)  17.13209  0.6443808
##  Ljung-Box Test     R^2  Q(10)  5.347538  0.8667866
##  Ljung-Box Test     R^2  Q(15)  9.029638  0.8759627
##  Ljung-Box Test     R^2  Q(20)  13.18302  0.8693941
##  LM Arch Test       R    TR^2   5.055834  0.956088
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -4.729448 -4.696775 -4.729565 -4.716650
```

Here again similar to the previous model, all the parameters of ARMA and GARCH are statistically significant.

d. Plot og conditional variances and the standardised residuals of the statistically significant models from above.

```r
par(mfrow=c(2,2))

# standardised residual we don't see much to any volatility

plot (fitg2@residuals/fitg2@sigma.t,type='l',ylab='standard residuals')

# plotting the conditional variances

plot (fitg2@sigma.t^2,type='l',ylab='conditional variances')

# standardised residual we don't see much to any volatility

plot (fitg3@residuals/fitg3@sigma.t,type='l',ylab='standard residuals')

# plotting the conditional variances
```

```
plot (fitg3@sigma.t^2,type='l',ylab='conditional variances')
```