

ISyE6402_HW4

Jim Liu

11/22/2020

Question 1: ARIMA(p,d,q) (15 Points)

With the growth rate data (first difference), use the iteration with the AIC metric selecting the minimum AIC (don't select a simpler model with higher AIC), select the best order of ARIMA model (max order 4,1,4).

Plot residual and square residual ACFs and interpret.

Perform and interpret tests for serial correlation and heteroscedasticity on model residuals.

```
orders <- orders[order(-orders$AIC),]  
tail(orders)
```

```
##      p d q      AIC  
## 12 1 0 0 -999.979  
## 36 3 0 4 -1000.316  
## 45 4 0 3 -1001.778  
## 2  0 0 0 -1001.952  
## 26 2 0 4 -1002.094  
## 44 4 0 2 -1003.803
```

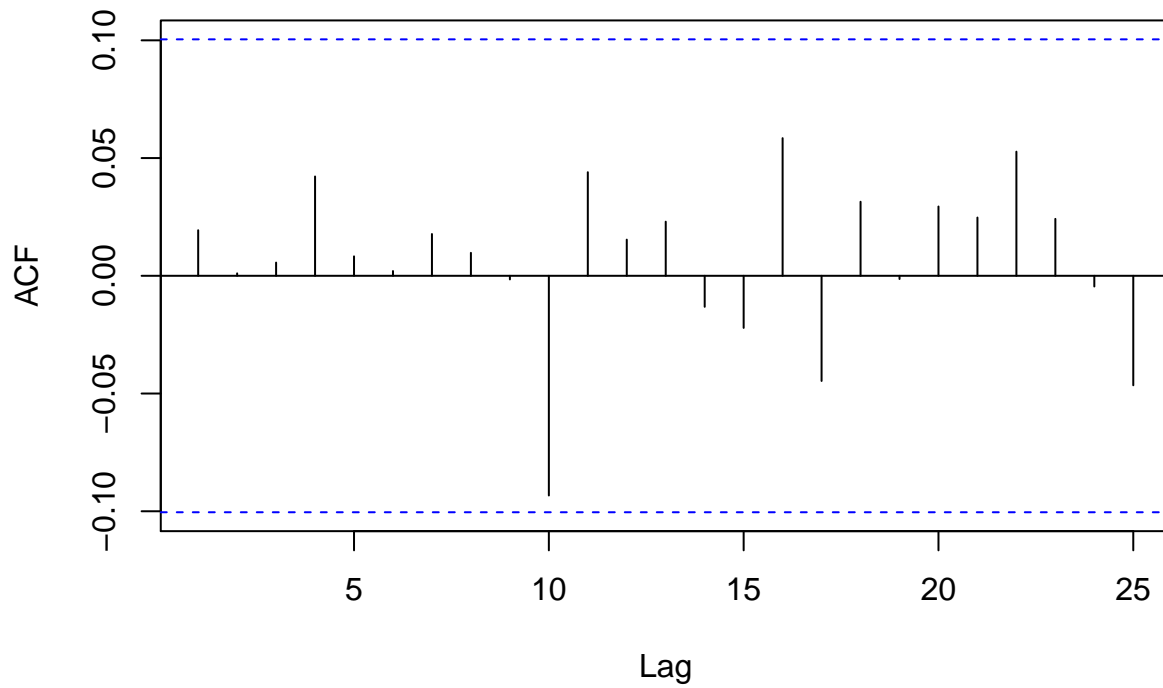
Based on AIC values, we select $p = 4, q = 2$ as our best ARMA model.

```
final.arima = arima(data.growth, order=c(4,0,2))  
final.arima
```

```
##  
## Call:  
## arima(x = data.growth, order = c(4, 0, 2))  
##  
## Coefficients:  
##          ar1      ar2      ar3      ar4      ma1      ma2  intercept  
##      -1.2937  -0.8501  0.1048  0.1193  1.3206  0.9363    0.0057  
## s.e.   0.0652   0.1153  0.0848  0.0568  0.0430  0.0840    0.0036  
##  
## sigma^2 estimated as 0.004049:  log likelihood = 509.9,  aic = -1005.8
```

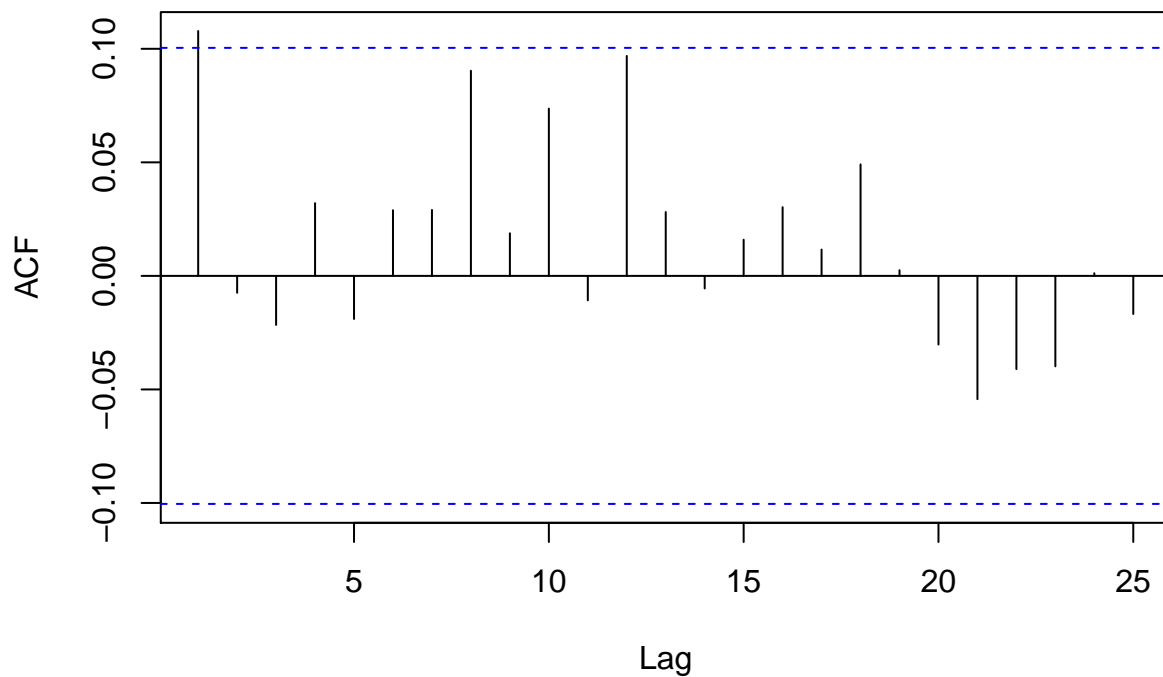
```
## Residual Analysis  
resids = resid(final.arima)[-1]  
acf(resids,main="Residuals of ARIMA Fit")
```

Residuals of ARIMA Fit



```
acf(resids^2,main="Squared Residuals of ARIMA Fit")
```

Squared Residuals of ARIMA Fit



From the two residual plots, they resemble white noise. Most of spikes are within the confidence bands. And, from the squared residuals, we don't see correlated residuals.

```

# for serial correlation
Box.test(resids,lag=7,type='Ljung',fitdf=6)

##
## Box-Ljung test
##
## data:  resids
## X-squared = 0.99634, df = 1, p-value = 0.3182

# for arch effect
Box.test((resids)^2,lag=7,type='Ljung',fitdf=6)

##
## Box-Ljung test
##
## data:  (resids)^2
## X-squared = 5.8578, df = 1, p-value = 0.01551

```

From Box-Ljung test on the residuals, we could not reject the null hypothesis, so we assume the residuals are uncorrelated, suggesting that the ARIMA model leads to uncorrelated residuals for this example. However, the p-value in the squared residuals are less than 0.5, so we reject the null hypothesis. And thus, we conclude that the squared residuals are correlated.

Question 2: ARMA(p,q)-GARCH(m,n) (20 Points)

With the growth rate data (first difference) and using the multi-step refinement method with minimum BIC starting with ARMA order calculated in Q1, find the best ARMA(P,Q)-GARCH(M,N) order pair. Fit this model.

Fully write out the equation. You can simply state the level of differencing Y represents in the data for simplicity sake.

Perform goodness of fit tests on this model and interpret. (Hint: summary of a garchFit model performs them all, you just need to interpret).

```

orders <- orders[order(-orders$BIC),]
tail(orders)

```

```

##      m n      BIC
## 2   0 0 -2.563903
## 17  2 3 -2.568767
## 12  1 4 -2.573193
## 21  3 1 -2.574947
## 10  1 2 -2.585301
## 11  1 3 -2.586141

```

From this GRACH order selection, we could see GRACH(1, 3) has smallest BIC values.

```

orders <- orders[order(-orders$BIC),]
tail(orders)

```

```
##      p q      BIC
## 17 2 3 -2.568748
## 3  0 1 -2.569097
## 8  1 0 -2.569152
## 29 4 3 -2.582689
## 2  0 0 -2.583971
## 28 4 2 -2.586141
```

With the GRACH selected order, we could see that ARMA(4, 2)-GARCH(1, 3) is our currently best model.

```
orders <- orders[order(-orders$BIC),]
tail(orders)
```

```
##      m n      BIC
## 2  0 0 -2.563903
## 17 2 3 -2.568767
## 12 1 4 -2.573193
## 21 3 1 -2.574947
## 10 1 2 -2.585301
## 11 1 3 -2.586141
```

Based on the updated order, our best model is still ARMA(4, 2)-GRACH(1, 3).

```
final.model.1 = garchFit(~ arma(4,2) + garch(1,3), data = data.growth, trace = FALSE)
```

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

```
## Warning: Using formula(x) is deprecated when x is a character vector of length > 1.
## Consider formula(paste(x, collapse = " ")) instead.
```

```
summary(final.model.1)
```

```
##
## Title:
## GARCH Modelling
##
## Call:
## garchFit(formula = ~arma(4, 2) + garch(1, 3), data = data.growth,
##          trace = FALSE)
##
## Mean and Variance Equation:
## data ~ arma(4, 2) + garch(1, 3)
## <environment: 0x7f887d470030>
## [data = data.growth]
##
## Conditional Distribution:
## norm
##
## Coefficient(s):
##      mu      ar1      ar2      ar3      ar4      ma1
## 0.00997019 -0.98371035 -0.48917816 0.08526319 0.12868712 0.99590186
```

```

##          ma2          omega          alpha1          beta1          beta2          beta3
## 0.55714910 0.00060916 0.09395281 0.00000001 0.15904150 0.59571041
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      9.970e-03 7.237e-03 1.378 0.168283
## ar1     -9.837e-01 1.289e-01 -7.630 2.35e-14 ***
## ar2     -4.892e-01 1.366e-01 -3.581 0.000343 ***
## ar3      8.526e-02 8.022e-02 1.063 0.287835
## ar4      1.287e-01 5.229e-02 2.461 0.013858 *
## ma1      9.959e-01 1.312e-01 7.591 3.18e-14 ***
## ma2      5.571e-01 1.580e-01 3.525 0.000423 ***
## omega    6.092e-04 3.500e-04 1.741 0.081769 .
## alpha1   9.395e-02 5.035e-02 1.866 0.062051 .
## beta1    1.000e-08      NA      NA      NA
## beta2    1.590e-01      NA      NA      NA
## beta3    5.957e-01      NA      NA      NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 519.0287      normalized: 1.358714
##
## Description:
## Sun Nov 22 19:52:10 2020 by user:
##
##
## Standardised Residuals Tests:
##
##      Statistic p-Value
## Jarque-Bera Test R Chi^2 80.57791 0
## Shapiro-Wilk Test R W 0.9788229 2.173299e-05
## Ljung-Box Test R Q(10) 7.269171 0.6998127
## Ljung-Box Test R Q(15) 8.710965 0.8921388
## Ljung-Box Test R Q(20) 10.26248 0.9631588
## Ljung-Box Test R^2 Q(10) 3.83082 0.9546564
## Ljung-Box Test R^2 Q(15) 5.553778 0.9863088
## Ljung-Box Test R^2 Q(20) 6.608869 0.9977845
## LM Arch Test R TR^2 5.217552 0.9503148
##
## Information Criterion Statistics:
##      AIC      BIC      SIC      HQIC
## -2.654601 -2.530661 -2.656495 -2.605431

```

Fully write out the equation.

$$Y_t = 0.00997 - 0.9837Y_{t-1} - 0.4892Y_{t-2} + 0.08526Y_{t-3} + 0.1287Y_{t-4} + Z_t + 0.9959Z_{t-1} + 0.5571Z_{t-2}, Z_t = \sigma_t R_t, \text{ where } R_t \sim iidN(0, 1)$$

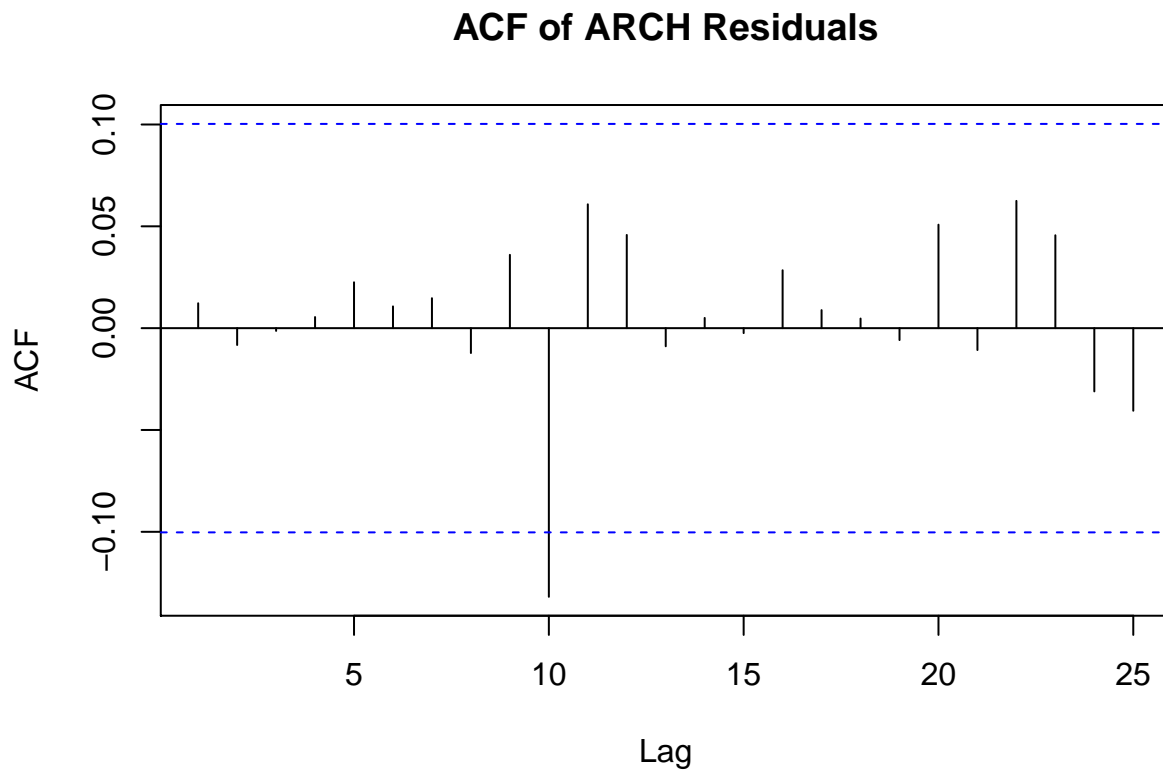
$$\sigma_t^2 = 0.0006092 + 0.09395Z_{t-1} + 0.00000001\sigma_{t-1}^2 + 0.159\sigma_{t-2}^2 + 0.5957\sigma_{t-3}^2$$

Perform goodness of fit tests on this model and interpret.

Based on Ljung-Box Tests on the residuals and squared residuals, those p-values are really high, so we could

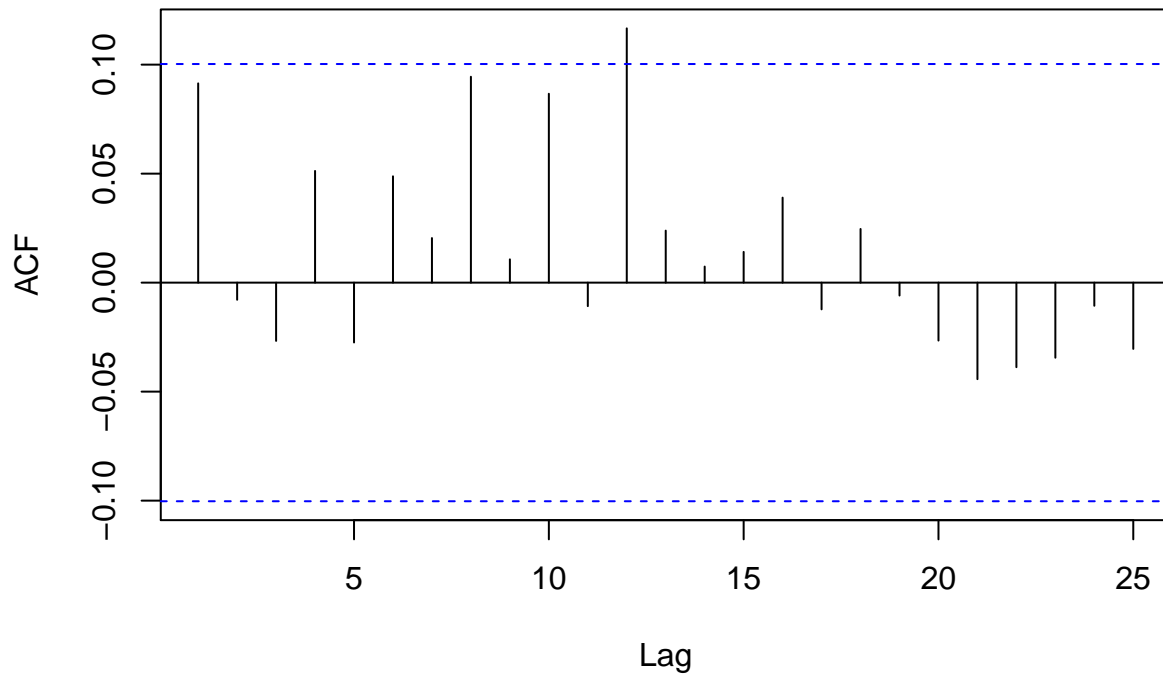
not reject the null hypothesis. Therefore, the residuals and squared residuals are uncorrelated.

```
resids.final.model = residuals(final.model.1)
acf(resids.final.model,main="ACF of ARCH Residuals")
```



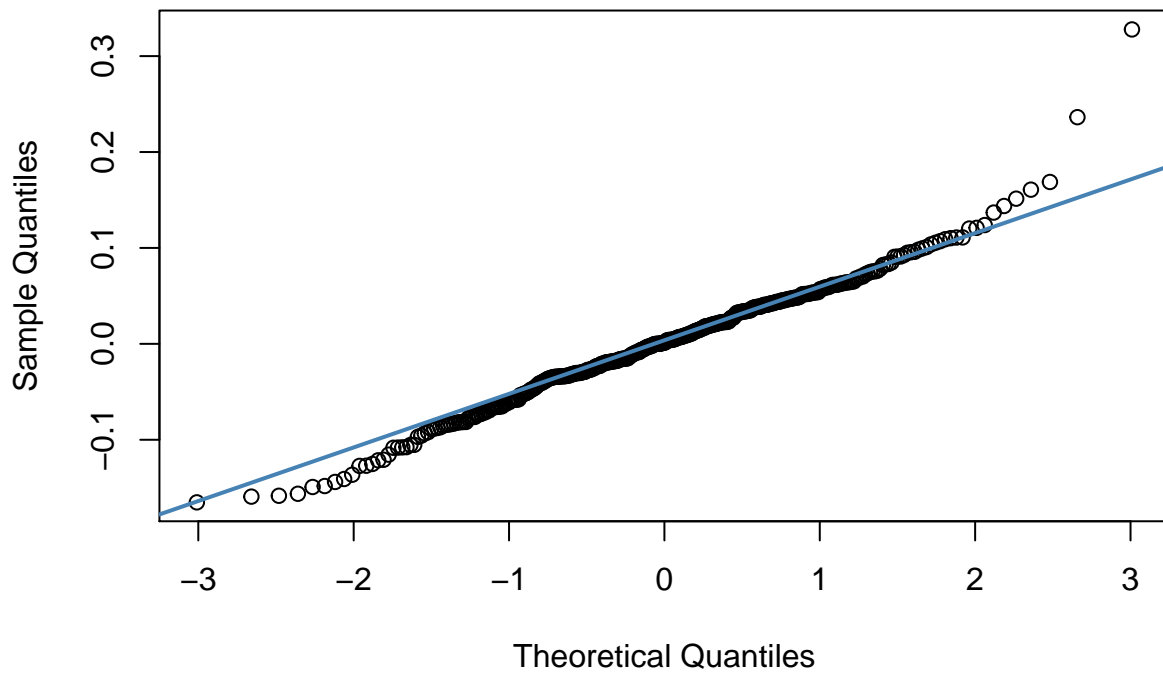
```
acf(resids.final.model^2,main="ACF of Squared ARCH Residuals")
```

ACF of Squared ARCH Residuals



```
qqnorm(resids.final.model)
qqline(resids.final.model, col = "steelblue", lwd = 2)
```

Normal Q-Q Plot



Based on the ACF plots and Q-Q plot, most spikes are within the confidence bands and the Q-Q plot seems not perfectly on straight line.

Question 3: Forecasting (15 Points)

Using the rolling forecasting method (for each pred, fit A-G model to all points prior), forecast last 40 points.

Calculate MAPE, and Precision

Overlay the draw an overlay of the predicted points on the original series in that range.

```
n=length(data.growth)
tlsa.test = data.growth[(n-39):n]
tlsa.train = data.growth[-c((n-39):n)]

## Prediction of the return time series
spec = ugarchspec(variance.model=list(garchOrder=c(1,3)),
                  mean.model=list(armaOrder=c(4, 2),
                                include.mean=T),distribution.model="std")
nfore = length(tlsa.test)
fore.series = NULL

for(f in 1: nfore){
  ## Fit models
  data = tlsa.train
  if(f>=2)
    data = c(tlsa.train,tlsa.test[1:(f-1)])

  final.model = ugarchfit(spec, data, solver = 'hybrid')

  ## Forecast
  fore = ugarchforecast(final.model, n.ahead=1)
  fore.series = c(fore.series, fore@forecast$seriesFor)
}
```

```
## Compute Accuracy Measures
### Mean Absolute Prediction Error (MAPE)
mean(abs(fore.series - tlsa.test)/abs(tlsa.test))
```

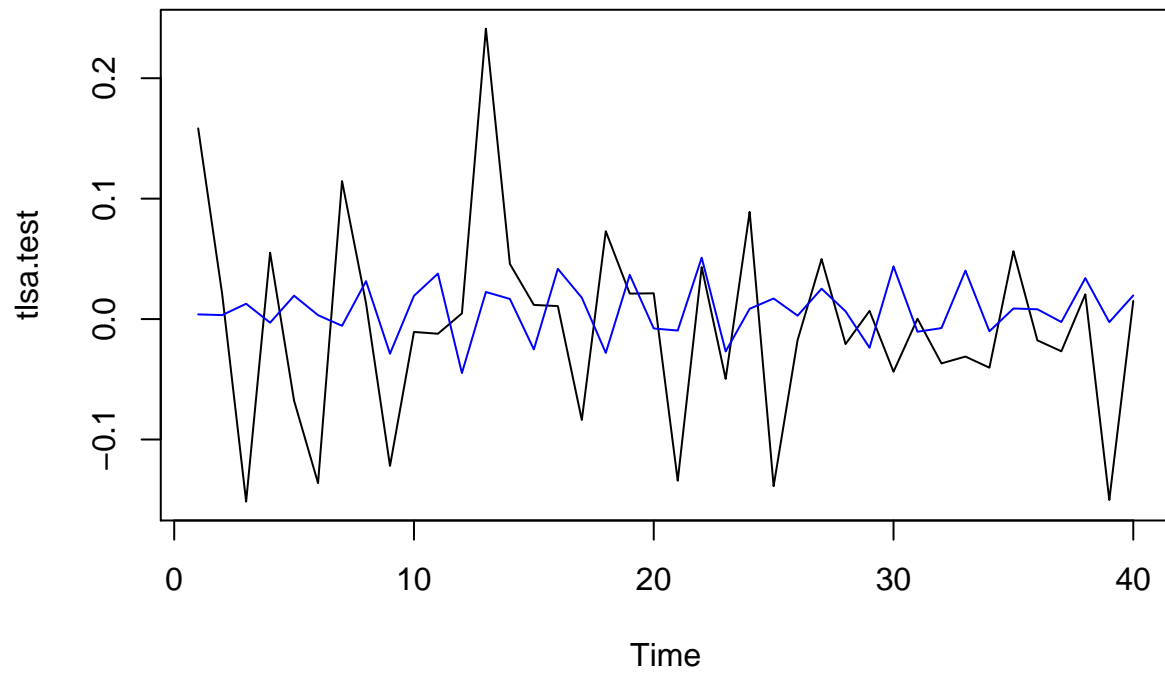
```
## [1] 2.52546
```

```
### Precision Measure (PM)
sum((fore.series - tlsa.test)^2)/sum((tlsa.test-mean(tlsa.test))^2)
```

```
## [1] 1.043868
```

Based on PM, we could see that the value is close to 1 which means two values has nearly the same spread of data points. As for MAPE, we could see that the value is not too small. The ideal situation is that the value close to zero which means the predictions are perfect. Therefore, the value 2.52546 does not meet perfectly the ideal situation.

```
par(mfrow=c(1,1))
ts.plot(tlsa.test)
lines(fore.series,col="blue")
```

From the overlayed plot, we could observe some points follow the trend in testing points and some don't.