Time Series Analysis on SNCF Traffic and UK Stock Index FTSE

Khanh TRUONG

December 2018

Time Series is a field of mathematical statistics which analyses data over a time period. While assumption of independence is usually used in classic statistics, time series deals with correlation of data in one time to another. This critical difference makes time series analysis have many specific methods compared to classical common statistics. Within the cope of this project, SARIMA and GARCH are going to be demonstrated via two data sets of SNCF Traffic and UK Stock Index FTSE. Note that this analysis aims to present the approaches by applying on real data set, backup definitions and theory of SARIMA and GARCH are supposed to be aware beforehand.

1 SNCF Traffic

In this part of the project, we analyze and predict the monthly SNCF traffic between 1963 and 1980, expressed in millions of travellers-kilometers. We are going to fit the models, study the remaining residuals and make prediction for the following year based on the model.

1.1 Fit models

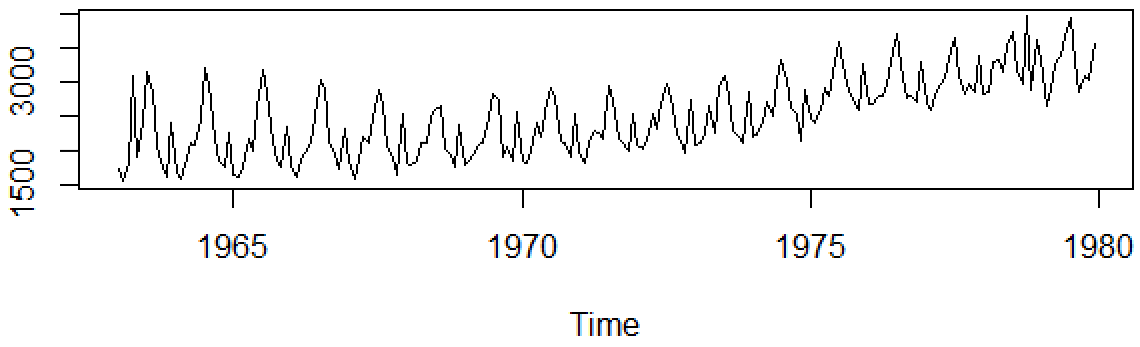

Figure 1: SNCF Traffic

First of all, a plot of original time series is always needed in order to have appropriate approach. It is obvious that there is some increasing trend and seasonality.

In order to remove the trend, we try differentiating the process $diffX_t = X_t - X_{t-1}$. The trend is removed but plot of diffX seems to still have seasonality.

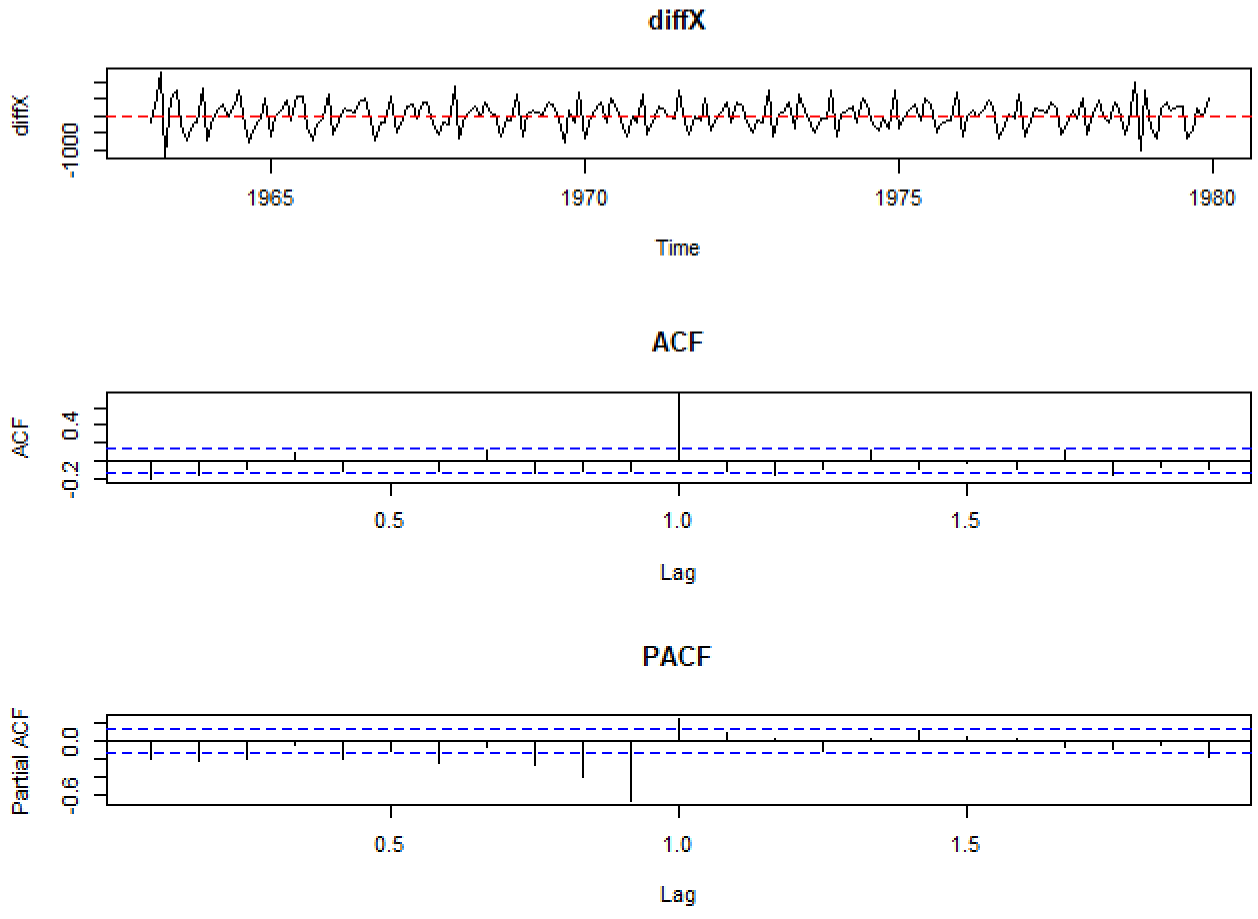

Figure 2: Difference X Order 1

ACF plot of $diffX$ pops up at lag 12 (lag 1.0 on the plot, since there are 12 observations in a year) confirms this seasonality. That gives a hint to try differentiating with lag 12.

$$diff12X = (I - B)^1 (I - B^{12})^1 X = (I - B) - (B^{12} - B^{13})X$$

$$diff12X_t = (X_t - X_{t-1}) - (X_{t-12} - X_{t-13})$$

Indeed, $diff12X$ gives much better stationary process for analysis, i.e. there is no trend and seasonality (Figure 3). In that manner, ACF vanishing after 13 lags indicates MA(13) to $diff12X$ may be a good start to fit the model (We can also specify beforehand which significant lags putting in our model - lag 1, 11, 12 and 13 for instance), or equivalently said we form $SARIMA(0, 1, 1)(0, 1, 1)_{12}$ to X.

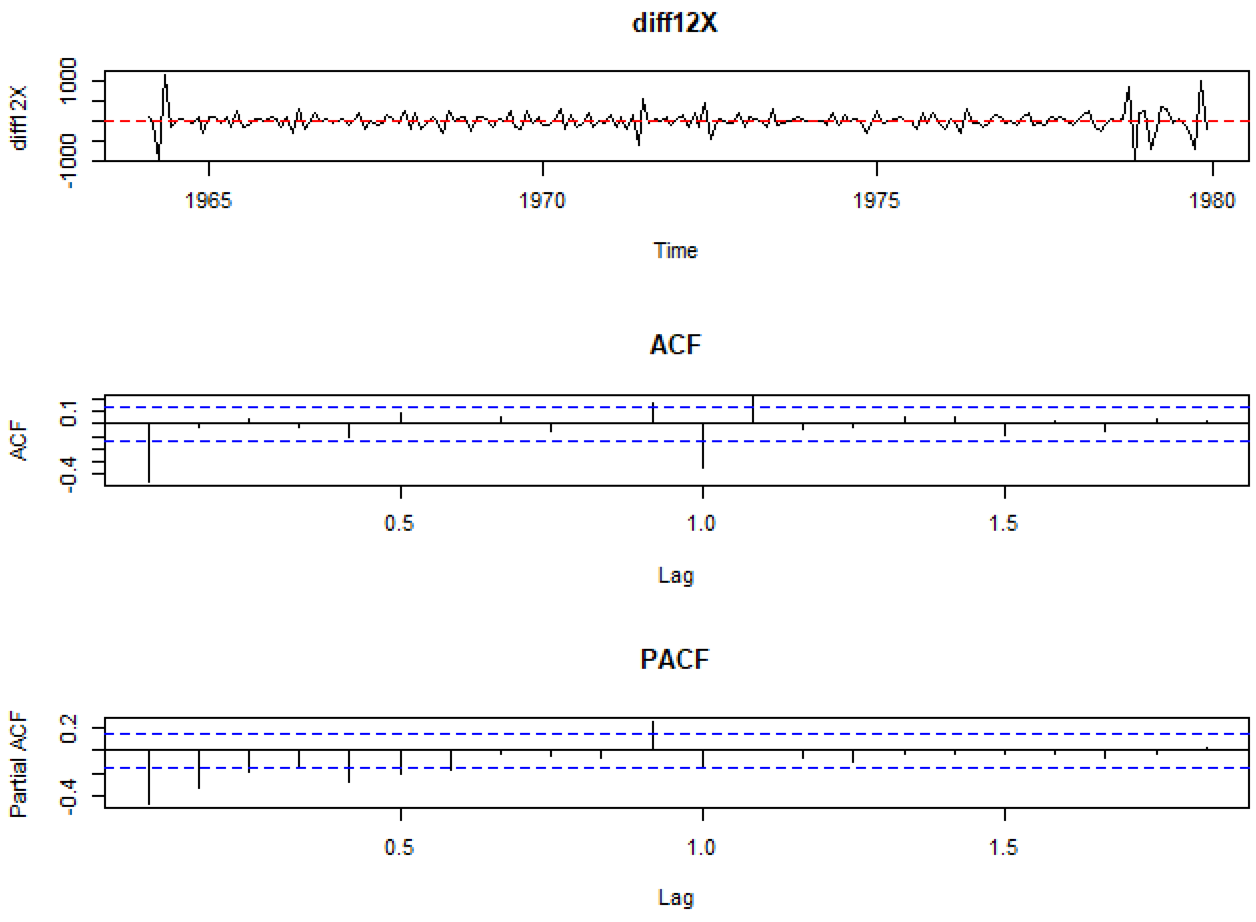

Figure 3: Difference X Order 1 Lag 12

```
> (sarima <- arima(X,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12)))

Call:
arima(x = X, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))

Coefficients:
          ma1     sma1
      -0.8725  -0.5387
s.e.   0.0412   0.0688

sigma^2 estimated as 26247:  log likelihood = -1245.64,  aic = 2497.28
```

The function *arima* in *R* generates estimated coefficients of the models. Accordingly, both coefficients (the ARMA part - *ma*1, and the seasonal part *sma*1) are significantly different from 0.

Alternatively, we can take advantage of function *auto.arima*. This function will run over multiple models with different coefficient values of SARIMA models then choose the one with smallest "BIC" criteria (we can choose other criteria such as "AIC").

```
> (sarima.auto <- auto.arima(X, trace=TRUE, ic='bic'))

 Fitting models using approximations to speed things up...

 ARIMA(2,1,2)(1,1,1)[12]                    : 2342.712
 ARIMA(0,1,0)(0,1,0)[12]                    : 2493.516
 ARIMA(1,1,0)(1,1,0)[12]                    : 2386.169
 ARIMA(0,1,1)(0,1,1)[12]                    : 2371.947
 ARIMA(2,1,2)(0,1,1)[12]                    : 2388.217
 ARIMA(2,1,2)(2,1,1)[12]                    : 2355.809
 ARIMA(2,1,2)(1,1,0)[12]                    : 2351.859
 ARIMA(2,1,2)(1,1,2)[12]                    : 2346.642
 ARIMA(2,1,2)(0,1,0)[12]                    : 2413.067
 ARIMA(2,1,2)(2,1,2)[12]                    : 2361.061
 ARIMA(1,1,2)(1,1,1)[12]                    : 2337.111
 ARIMA(1,1,1)(1,1,1)[12]                    : 2332.73
 ARIMA(0,1,0)(1,1,1)[12]                    : 2401.564
 ARIMA(1,1,1)(0,1,1)[12]                    : 2378.836
 ARIMA(1,1,1)(2,1,1)[12]                    : 2343.995
 ARIMA(1,1,1)(1,1,0)[12]                    : 2342.096
 ARIMA(1,1,1)(1,1,2)[12]                    : 2336.542
 ARIMA(1,1,1)(0,1,0)[12]                    : 2403.357
 ARIMA(1,1,1)(2,1,2)[12]                    : 2349.247
 ARIMA(0,1,1)(1,1,1)[12]                    : 2329.014
 ARIMA(0,1,2)(1,1,1)[12]                    : 2330.563
 ARIMA(0,1,1)(2,1,1)[12]                    : 2345.323
 ARIMA(0,1,1)(1,1,0)[12]                    : 2337.881
 ARIMA(0,1,1)(1,1,2)[12]                    : 2332.037
 ARIMA(0,1,1)(0,1,0)[12]                    : 2397.711
 ARIMA(0,1,1)(2,1,2)[12]                    : 2350.148

 Now re-fitting the best model(s) without approximations...

 ARIMA(0,1,1)(1,1,1)[12]                    : 2510.109

 Best model: ARIMA(0,1,1)(1,1,1)[12]

Series: X
ARIMA(0,1,1)(1,1,1)[12]

Coefficients:
          ma1     sar1     sma1
      -0.8737  -0.2734  -0.3348
s.e.   0.0415   0.1767   0.1638

sigma^2 estimated as 26311:  log likelihood=-1244.55
AIC=2497.1   AICc=2497.31   BIC=2510.11
```

The model with the lowest "BIC" criteria is $SARIMA(0,1,1)(1,1,1)_{12}$. The coefficient $sar1$ is, however, not significantly different from 0 (confidence interval contains 0, or equivalently said p-value is greater than 0.05). Therefore, we come back to the same above proposed model which is based on the plots.

```
> confint(sarima.auto, level=0.95) # CI of sar1 contains 0
          2.5 %      97.5 %
ma1  -0.9549785 -0.79243629
sar1 -0.6197861  0.07288717
sma1 -0.6558314 -0.01380554
> t.statistic <- sarima.auto$coef['sar1']/ # estimate coefficient
+   sqrt(sarima.auto$var.coef['sar1','sar1']) # standard error
> pt(abs(t.statistic), df=1, lower.tail=FALSE)*2 # p-value
     sar1
0.3652325
```

1.2 Study residuals

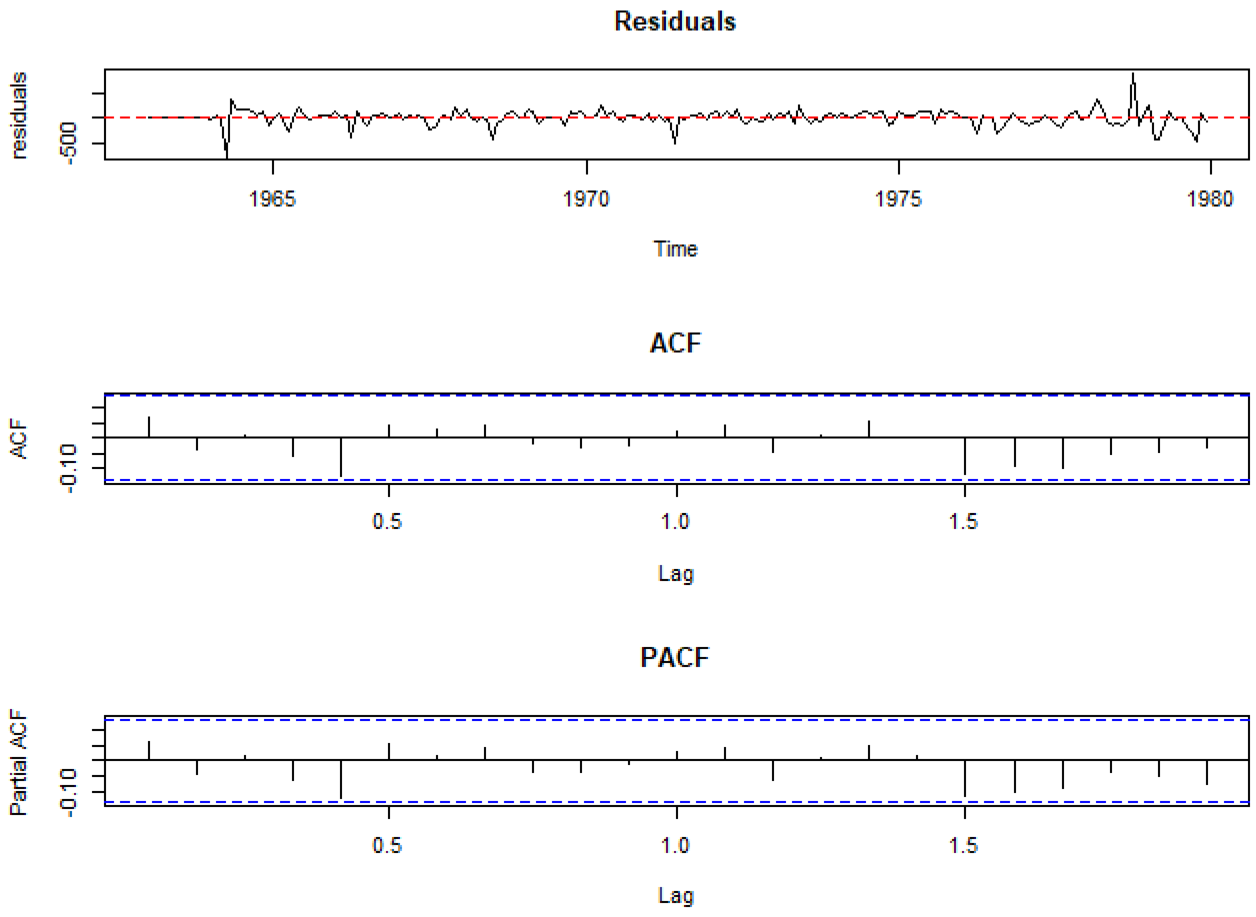

Figure 4: SARIMA residuals

After fitting a good time series model, the remaining residuals should be a (weak) white noise. We are going to see if this true in our case. The plot of residuals, its acf and pacf in Figure 4 prove no visual correlation. Ljung-Box test confirms this by obtaining very large p-value. (Ljung-Box is a test to see if a time series has auto-correlation. Null hypothesis is there is no serial correlation upto n lags. In our case, we select option *lag* = 12, corresponding to 1 year period; but different near-by values, such as 6 or 24, will not change the insignificant result in our problem).

```
> Box.test(residuals, lag=12, type='Ljung-Box')

        Box-Ljung test
data:  residuals
X-squared = 6.8145, df = 12, p-value = 0.8696
```

It is good that the there is no correlation in the residuals. Nevertheless, we need to examine if the volatility of the residuals is stable during the period. In order to conduct the test, we only need to replace residuals in the above Ljung-Box test by squared residuals. Null hypothesis is that there is no heteroskedastic effect. Luckily we can not reject the null hypothesis with p-value = 0.3455 (if we rejected, we should have applied model GARCH - which will be presented in the second part of the project). Therefore, results from the SARIMA model is good enough.

```
> Box.test(residuals^2, lag=12, type='Ljung-Box')

        Box-Ljung test
data:  residuals^2
X-squared = 13.33, df = 12, p-value = 0.3455
```

Although residuals are generally not required to be normal distributed, we would like to see quickly the shape of the histogram of the residuals and how it differs from Gaussian distribution.

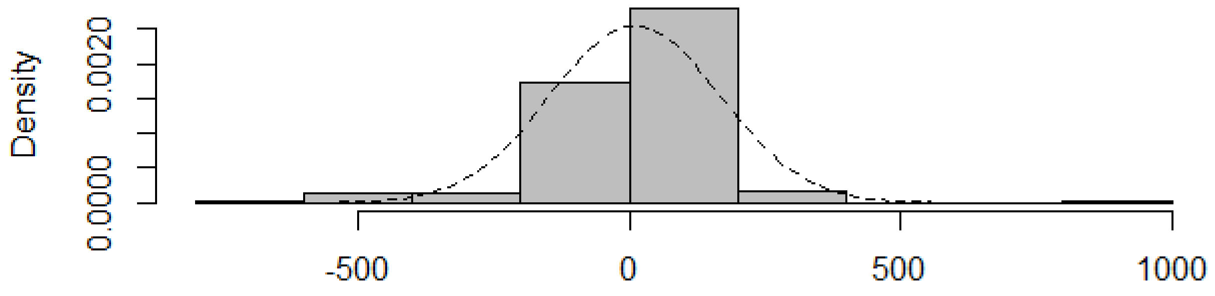

Figure 5: SARIMA Residuals Distribution

```
> jarque.bera.test(residuals)

        Jarque Bera Test
data:  residuals
X-squared = 607.47, df = 2, p-value < 2.2e-16
```

The residuals seem to concentrate greatly to the middle and, therefore, have slight flat tails (Figure 5). This result is probably even better than Gaussian, since the error of model (residuals are literally the differences between actual values and fitted values) are more close to 0. As expected, negligible p-value in Jarque–Bera test confirms that residuals follow some distribution but not Gaussian (Jarque–Bera test is a goodness-of-fit test to see if a data matches Gaussian distribution. Null hypothesis is that data follows normal distribution).

1.3 Forecast

After fitting SARIMA and obtaining satisfactory residuals, we are going to conduct prediction based on the model. In fact, the data used to train the model is excluded the last year - 1980, from original data set. This gives us the opportunity to compare how close the actual values and predicted values.

```
> (forecast <- forecast(sarima, h=12))
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 1980       3150.905 2943.282 3358.528 2833.374 3468.436
Feb 1980       2833.319 2624.015 3042.624 2513.216 3153.423
Mar 1980       3102.293 2891.320 3313.266 2779.638 3424.949
Apr 1980       3342.804 3130.175 3555.432 3017.616 3667.991
May 1980       3359.275 3145.004 3573.546 3031.575 3686.974
Jun 1980       3714.140 3498.239 3930.042 3383.948 4044.333
Jul 1980       3959.685 3742.165 4177.204 3627.018 4292.352
Aug 1980       3366.723 3147.598 3585.848 3031.600 3701.846
Sep 1980       3030.824 2810.104 3251.543 2693.262 3368.385
Oct 1980       3391.165 3168.863 3613.468 3051.183 3731.148
Nov 1980       3072.033 2848.159 3295.907 2729.647 3414.419
Dec 1980       3649.556 3424.121 3874.991 3304.783 3994.329
```

Function *forecast* in package *forecast* in R will produce the predicted values given model. It will also generate 0.80 and 0.95 confidence intervals. We plot this result compared to actual values in 1980 (Figure 6).

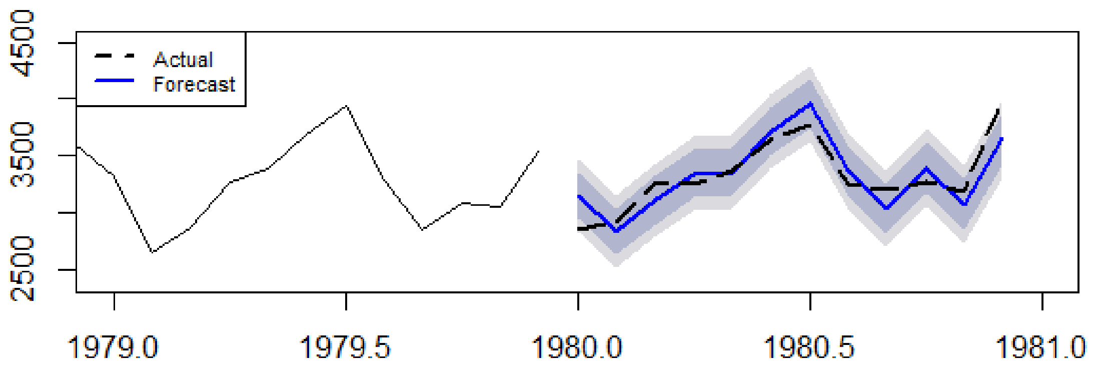

Figure 6: SARIMA Forecast

The predicted values seem to be very closed to actual ones. Moreover, the actual values are always in the predicted 0.80 confidence intervals (dark grey area). Nevertheless, the actual values are more stable, while the model forecasts they should be more fluctuated (forecasted peak is higher than actual peak, forecasted lowest level is lower than the actual's lowest).

1.4 Compare SARIMA to "decompose" method

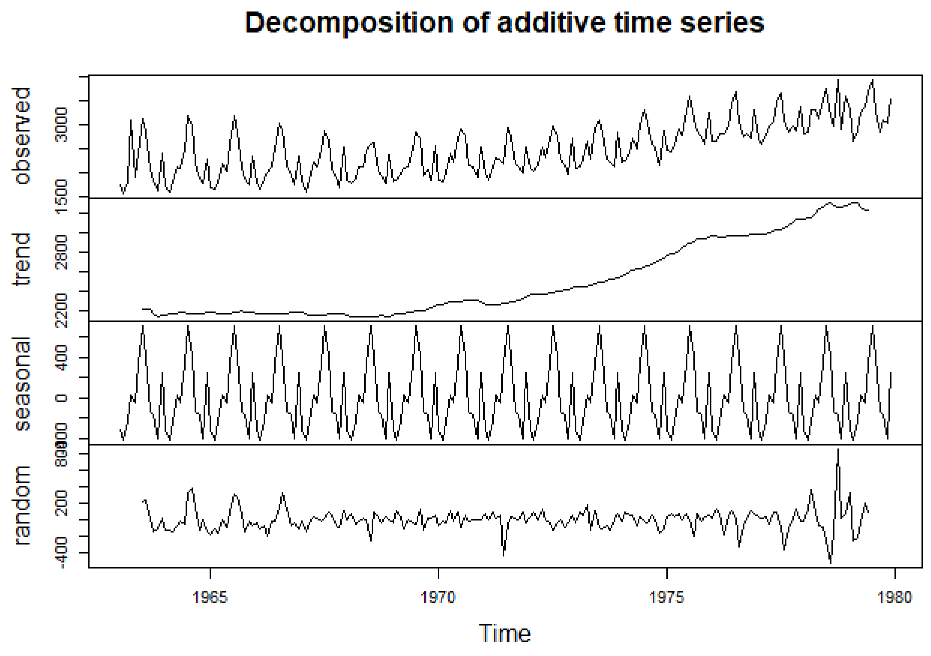

Figure 7: Decompose SNCF Traffic

A time series is usually considered as a combination of trend series, seasonality series and remaining residuals (Figure 7). Function *decompose* in R will split the original data into the three components (in this project's plots, residuals of *decompose* method is notated by "random", to distinguish to "residuals" of SARIMA method).

$$Y_t = trend_t + seasonality_t + residual_t$$

The plots residuals from the two approaches are fairly alike (Figure 8). However, acf and pacf plots (Figure 9 and 10) clearly manifest the superior of SARIMA when there is no correlation between lags (as discuss in part 1.2), while there are numerous lags out of confidence internal 0 in method *decompose*. We also reject the null hypothesis with trivial p-value in Ljung-Box test. So the residuals of *decompose* is not yet a (weak) white noise.

```
> Box.test(random, lag=12, type='Ljung-Box')

        Box-Ljung test

data:  random
X-squared = 74.49, df = 12, p-value = 4.587e-11
```

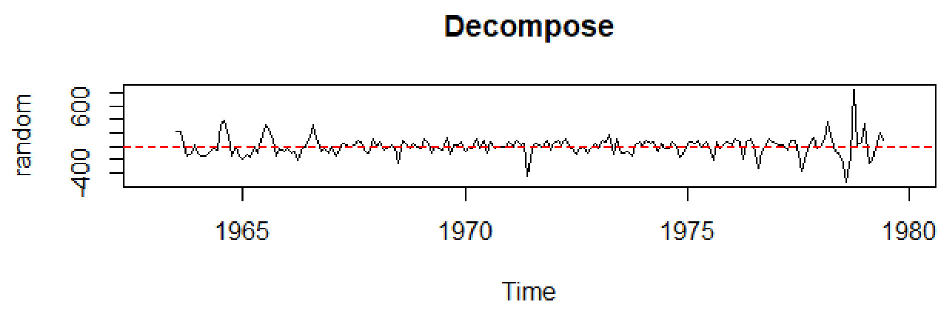

Figure 8: Residials of Decompose and SARIMA

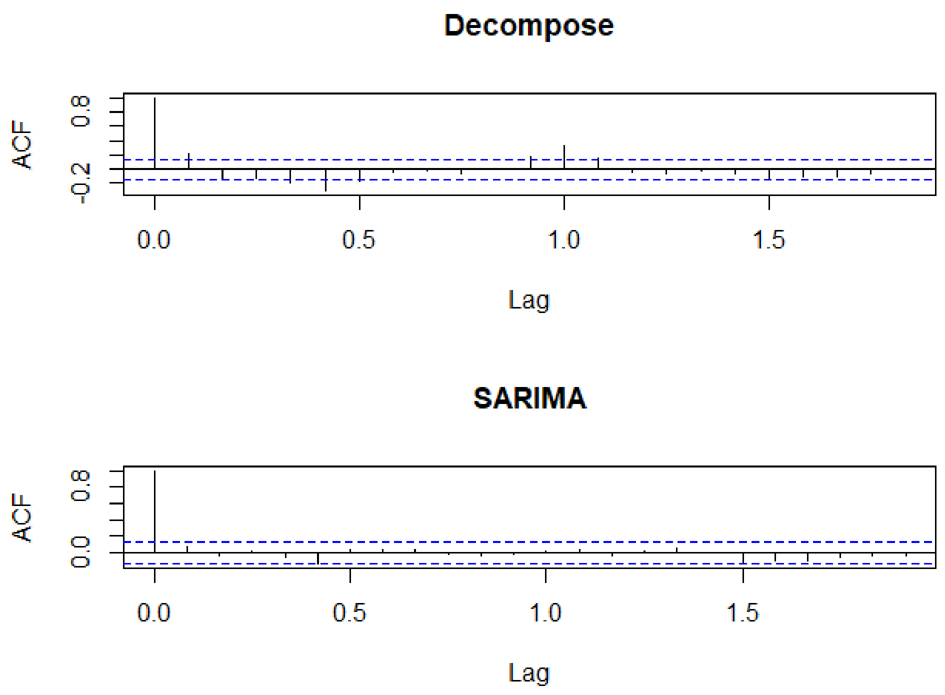

Figure 9: Residuals ACF of Decompose and SARIMA

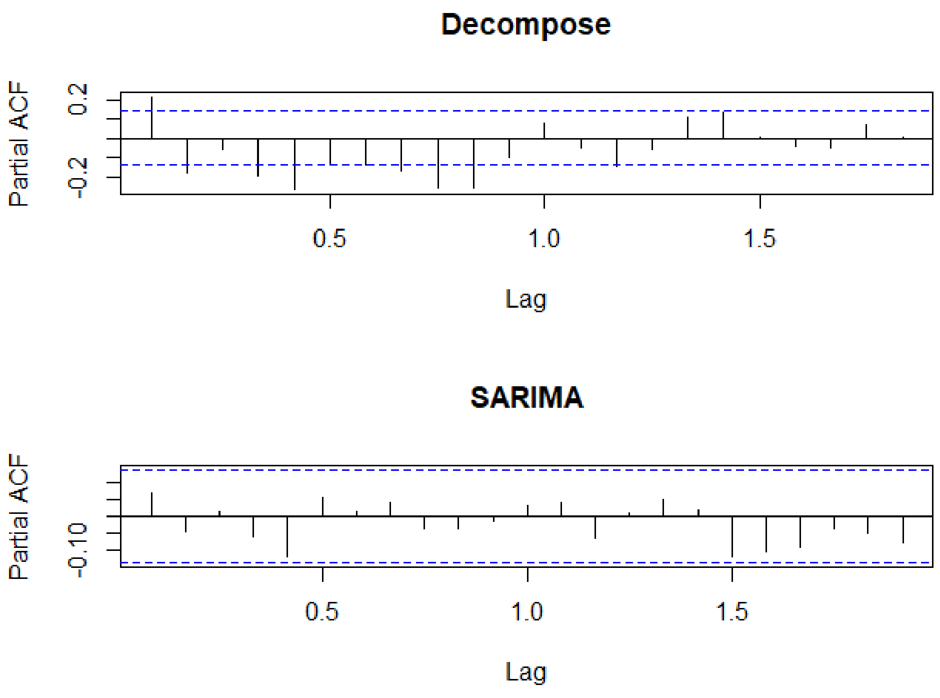

Figure 10: Residuals PACF of Decompose and SARIMA

We also have a brief look at the shape of *decompose*'s residuals. Similar to the one in SARIMA, the histogram is symmetric at 0 but has slight tails, which is significant different from Gaussian distribution (Figure 11, and result from Jarque-Bera test).

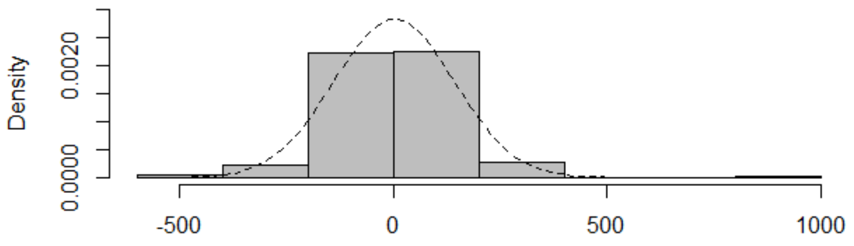

Figure 11: Decompose Residual Distribution

Despite of correlated residuals, the prediction of decompose method is not that bad, especially at the first and last quarters of 1980 (Figure 12). Nonetheless, SARIMA still foresees the future better. Besides, decompose seems to over-estimate the whole year 1980 (the red line is always above the dot black line).

Note that original function *decompose* in R does not allow to make prediction, since the extracted trend is literally the arithmetic mean of the time series (as a result, we can not obtain arithmetic mean in the future, even neither period at the beginning and at the end of time series window). Thus, we replace the arithmetic

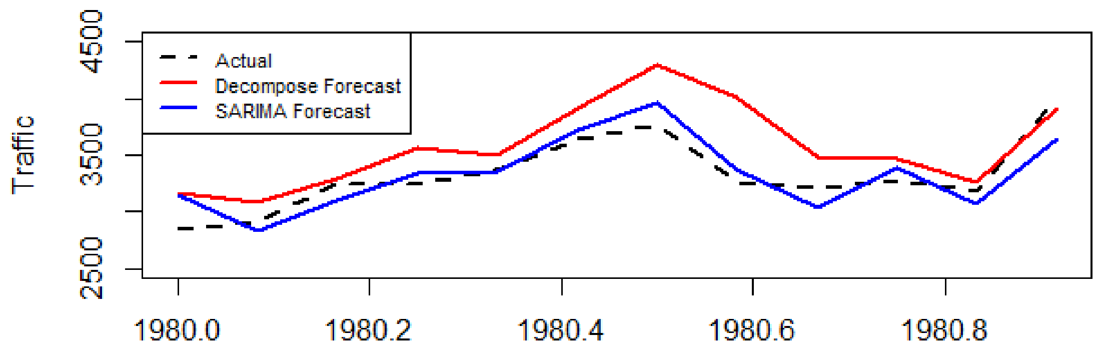

Figure 12: Decompose and SARIMA Forecast

mean by fitted values in regression model (regress over time) to represent the trend. (See more on full code script appendix)

$$trend = T + T^2 + error$$

$$forecast_{decompose} = trend_{regression} + seasonality_{decompose}$$

We have also tried another form of *decompose* which is “multiplicative”, instead of “additive” as presented. The behaviour of remaining residuals and the prediction is fairly similar to “additive” method. Therefore, SARIMA is out-performed over both additive decompose and multiplicative decompose.

$$Y_t = trend_t * seasonality_t * residual_t$$

1.5 Validate SARIMA model

Remind that we have not made use of data 1980 in our model (the last year data is used only for prediction comparison purpose). Therefore, in order to acquire the final model which takes advantage of all available data, we should apply SARIMA model again to the whole data set including 1980.

```
> (sarima <- arima(Y, order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12)))

Call:
arima(x = Y, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))
Coefficients:
          ma1     sma1
      -0.8721  -0.5698
s.e.   0.0391   0.0623
sigma^2 estimated as 26561:  log likelihood = -1325.23,  aic = 2656.47
```

To sum up, SARIMA model behaves very well to the data of SNCF. It absolutely can be applied in practice where managers of the company could make decisions and strategies based on the model’s predicted traffic.

2 ARCH and GARCH Process

In this section, we will simulate ARCH and GARCH processes and study their characteristics. The results of this section will be helpful to investigate financial series in part 3.

2.1 ARCH Process

We simulate an ARCH process with two coefficients 0.05 and 0.3:

$$\sigma_t{}^2 = 0.05 + 0.3X_{t-1}^2$$

```
> set.seed(1234)
> arch <- garch.sim(n=1000,alpha = c(0.05,0.3))
```

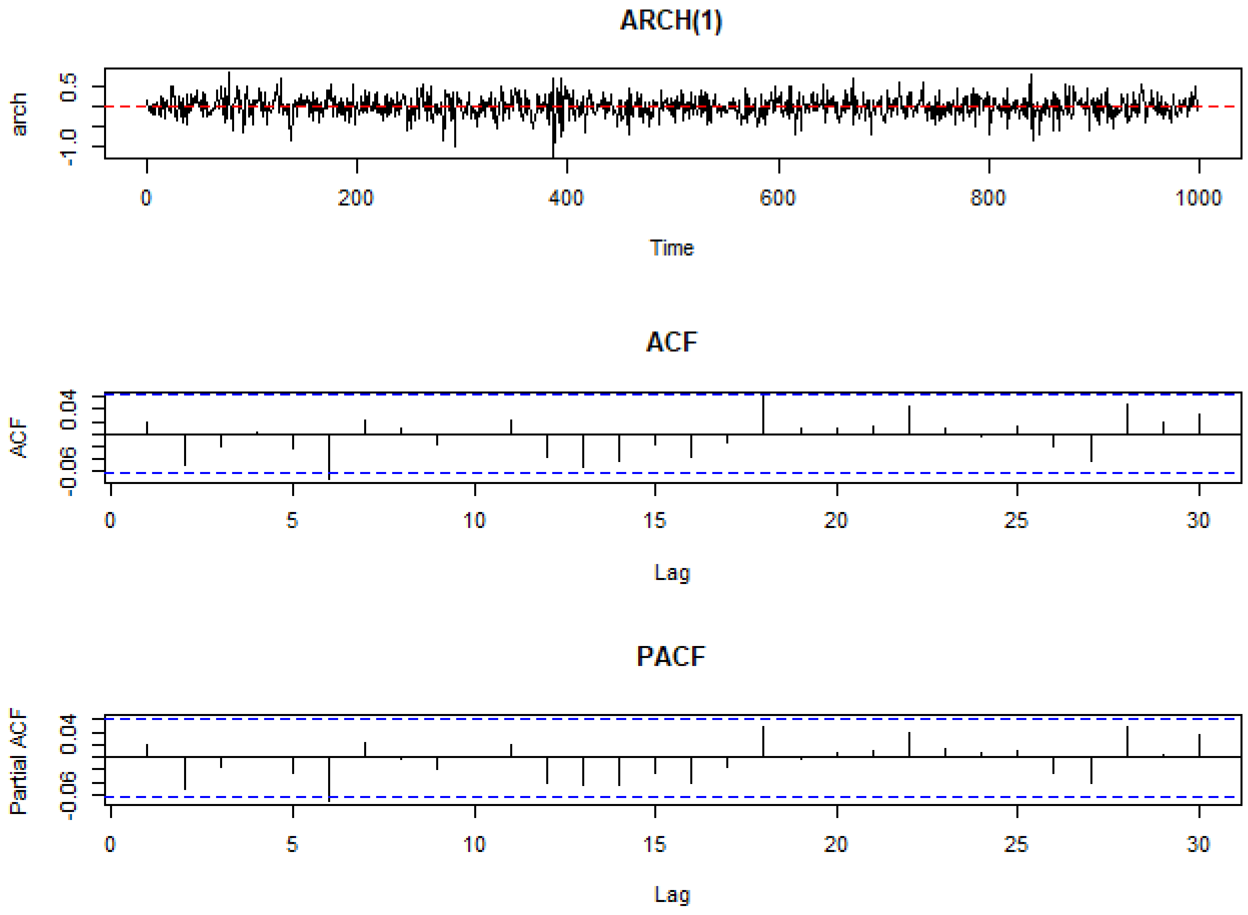

Figure 13: Simulated ARCH Process

As depicted in acf and pacf plots of ARCH(1), there are some lags that approach the confidence intervals round 0 but they are ambiguous (Figure 13). P-value > 0.05 in Ljung-Box test supports us to not reject the null hypothesis (recall null hypethesis in Ljung-Box test is that there is no serial correlation upto n lags). So the series looks to be (weak) white noise.

```
> Box.test(arch, lag=20, type='Ljung-Box')

        Box-Ljung test

data:  arch
X-squared = 21.553, df = 20, p-value = 0.3652
```

Theoretically, if

$$Var(y_t|y_{t-1}) = \sigma_t^2 = 0.05 + 0.3X_{t-1}^2$$

then y_t^2 should follow $AR(1)$ model, i.e.

$$y_t^2 = 0.05 + 0.3y_{t-1}^2 + \epsilon_t$$

This theoretical result is confirmed by pacf plot of squared value of the series, where lag 1 pops up.

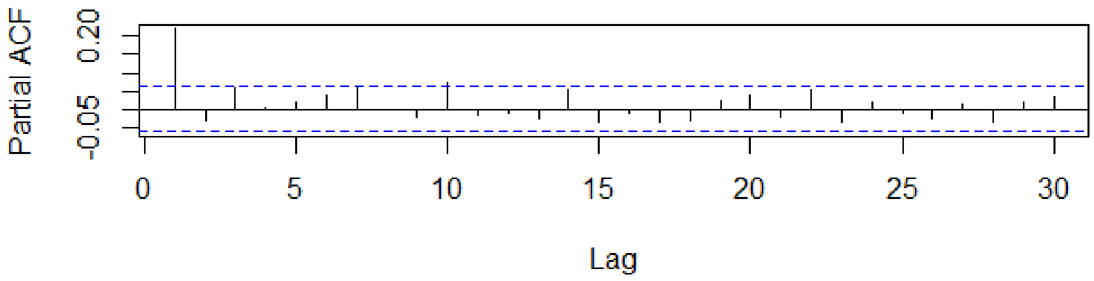

Figure 14: PACF of Squared ARCH

When we replace the process by its squared value to test if there is heteroskedastic effect, the negligible p-value gives significant evidence to reject the null hypothesis (null hypothesis is that there is no heteroskedastic effect), i.e. the volatility of the process is not identical during the time window.

```
> Box.test(arch^2, lag=20, type='Ljung-Box')

        Box-Ljung test

data:  arch^2
X-squared = 73.818, df = 20, p-value = 4.283e-08
```

2.2 GARCH Process

Applying similar procedure of ARCH, we simulate an GARCH(1,2) process with two coefficients 0.05 and 0.3 of ARCH part (including the constant term) and two coefficients 0.1 and 0.15 of GARCH part:

$${\sigma_t}^2 = 0.05 + 0.3X_{t-1}^2 + 0.1\sigma_{t-1}^2 + 0.15\sigma_{t-2}^2$$

```
> set.seed(1234)
> garch <- garch.sim(n=1000, alpha = c(0.05,0.3), beta=c(0.1,0.15))
```

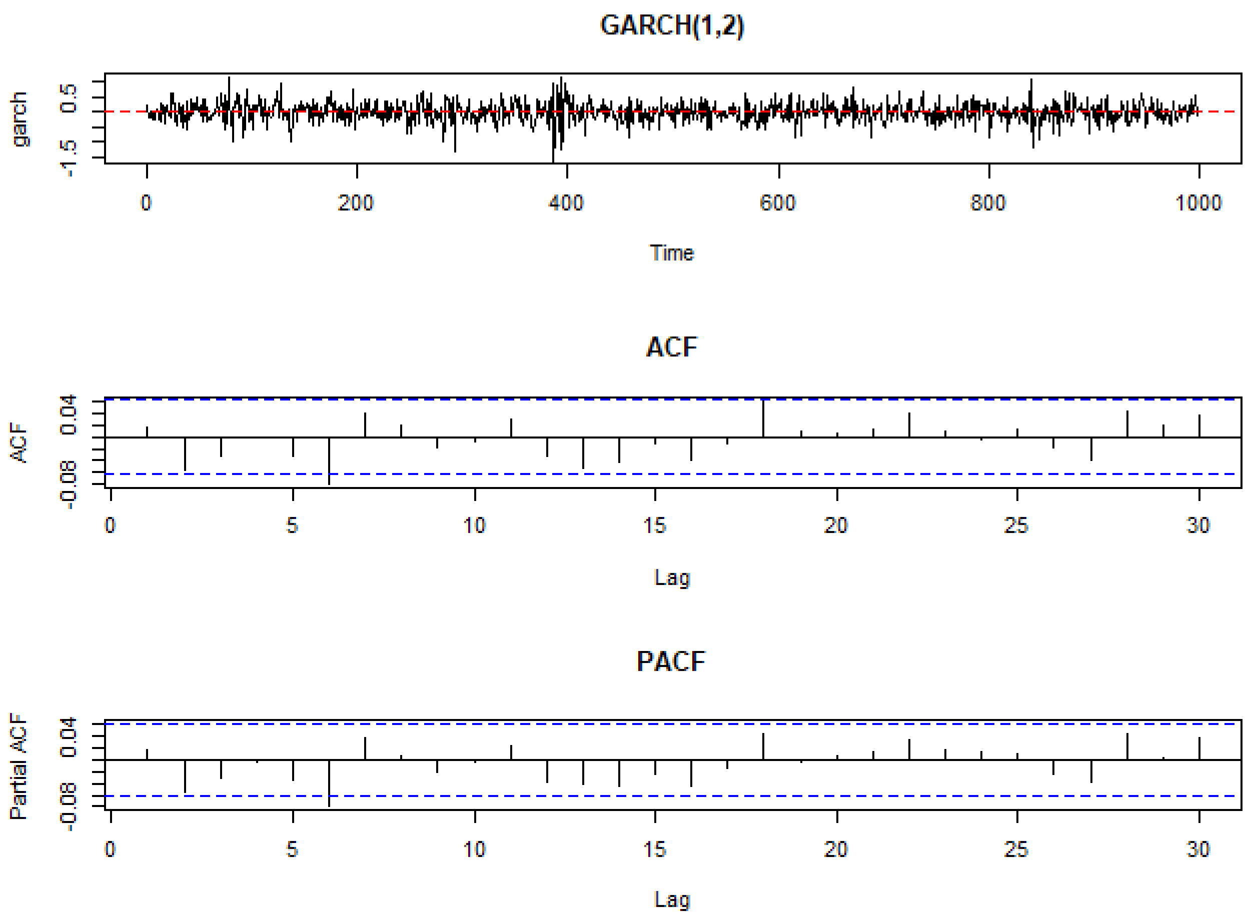

Figure 15: Simulated GARCH Process

The plot of the process, its acf and pacf (Figure 15), as well as the Ljung-Box test demonstrate GARCH seems to be a (weak) white noise.

Moreover, Ljung-Box test for the squared value proves that the volatility of the series is not stable.

```
> Box.test(garch^2, lag=20, type='Ljung-Box')

        Box-Ljung test

data:  garch^2
X-squared = 196.52, df = 20, p-value < 2.2e-16
```

Theoretically, the squared GARCH y_t^2 process should follow some ARMA model. This fact is verified by numerous significant lags in acf and pacf plots of squared GARCH y_t^2 (Figure 16).

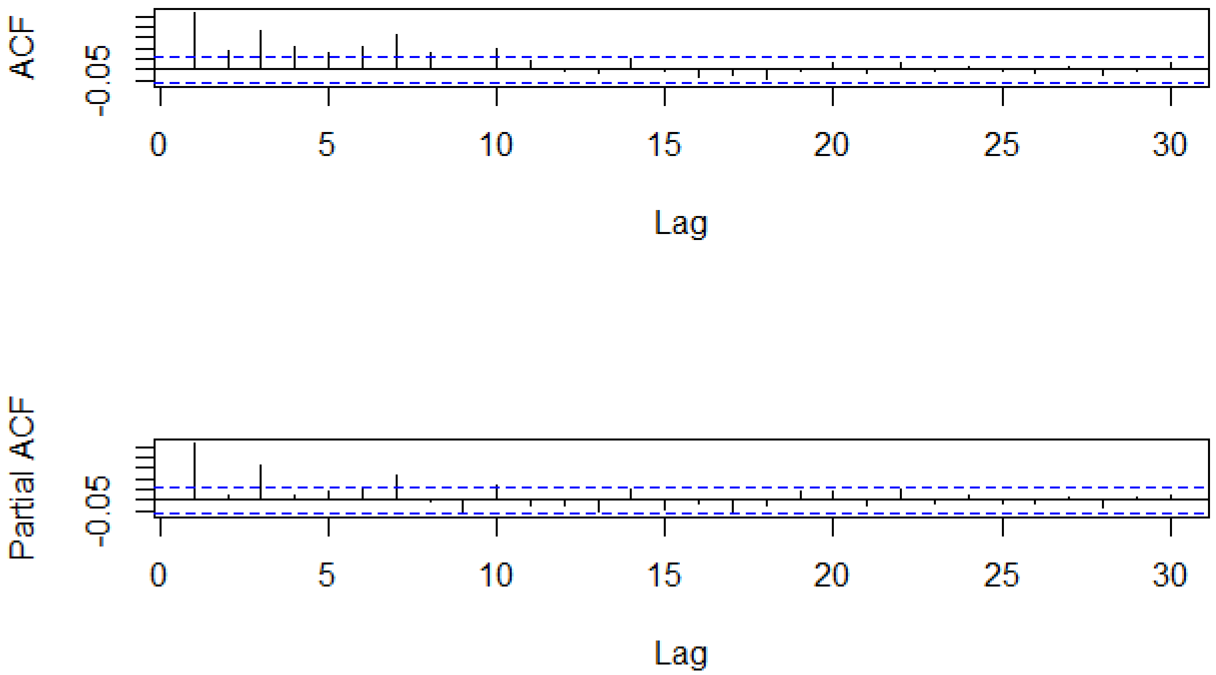

Figure 16: ACF and PACF of Squared GARCH

In conclusion, both ARCH and GARCH processes are visually (weak) white noise where conditional variances are not constant over the time window. Moreover, the squared values follow AR and ARMA models, respectively.

3 UK Stock Index FTSE

Found in 1571, London Stock Exchange (LSE) is one of the world's oldest stock exchanges. The market capitalisation of LSE was $4.38 trillion in March 2018 (1). The share index of the 100 companies listed on LSE with the highest market capitalisation form into The Financial Times Stock Exchange 100 Index, also called the FTSE. This index reflects the general movement of the market exchange and is too important so that may affect not only the UK economy but the global.

We are going to analyze FTSE from data "EuStockMarkets" in *R* which comprises daily index from middle 1991 to middle 1998. Similarly to the first part of this project, we are going to fit the models, study the remaining residuals and make prediction for the next two weeks based on the model.

3.1 Fit model

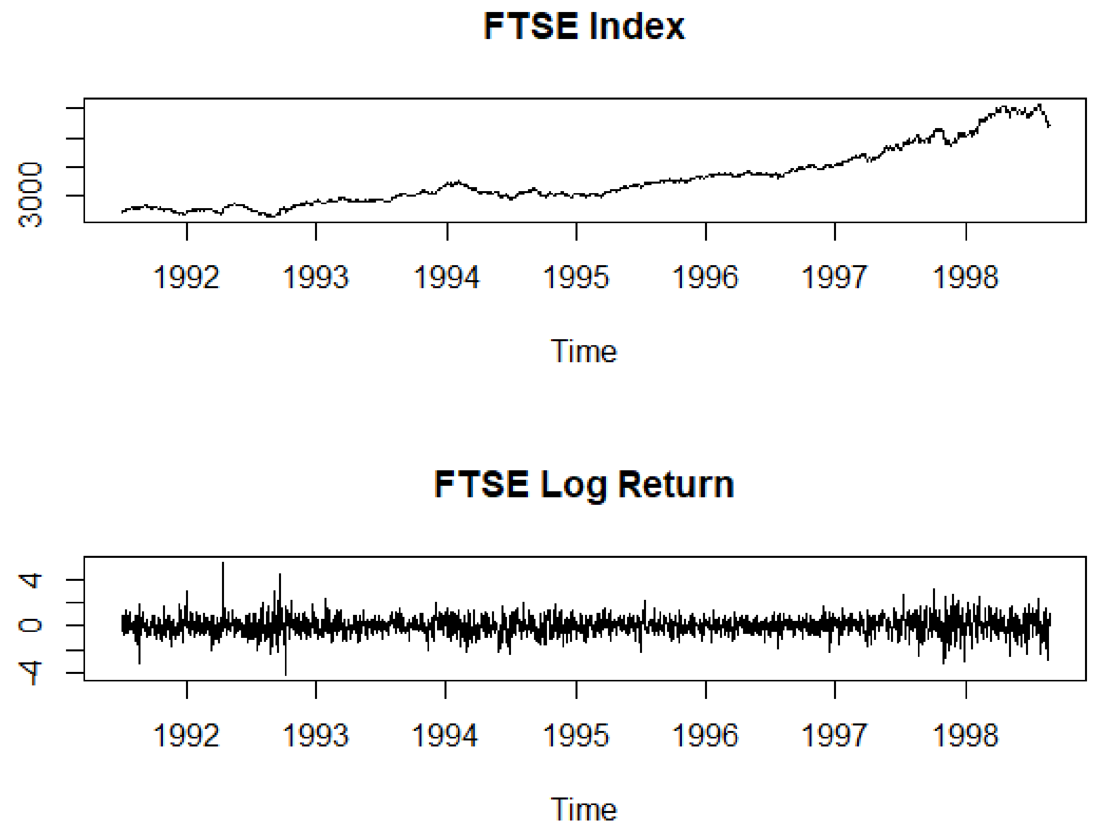

Figure 17: FTSE

As nature of many financial time series, FTSE seems to be rising during the time period. In order to remove the trend, we calculate the log-return of the index. The transformed series is now ready for analysis (Figure 17). From now on, we denote the log-return FTSE as "FTSE" for simplicity.

$$logreturn_t = 100 * [log(Index_t) - log(Index_{t-1})] = 100 * (log(\frac{Index_t}{Index_{t-1}}))$$

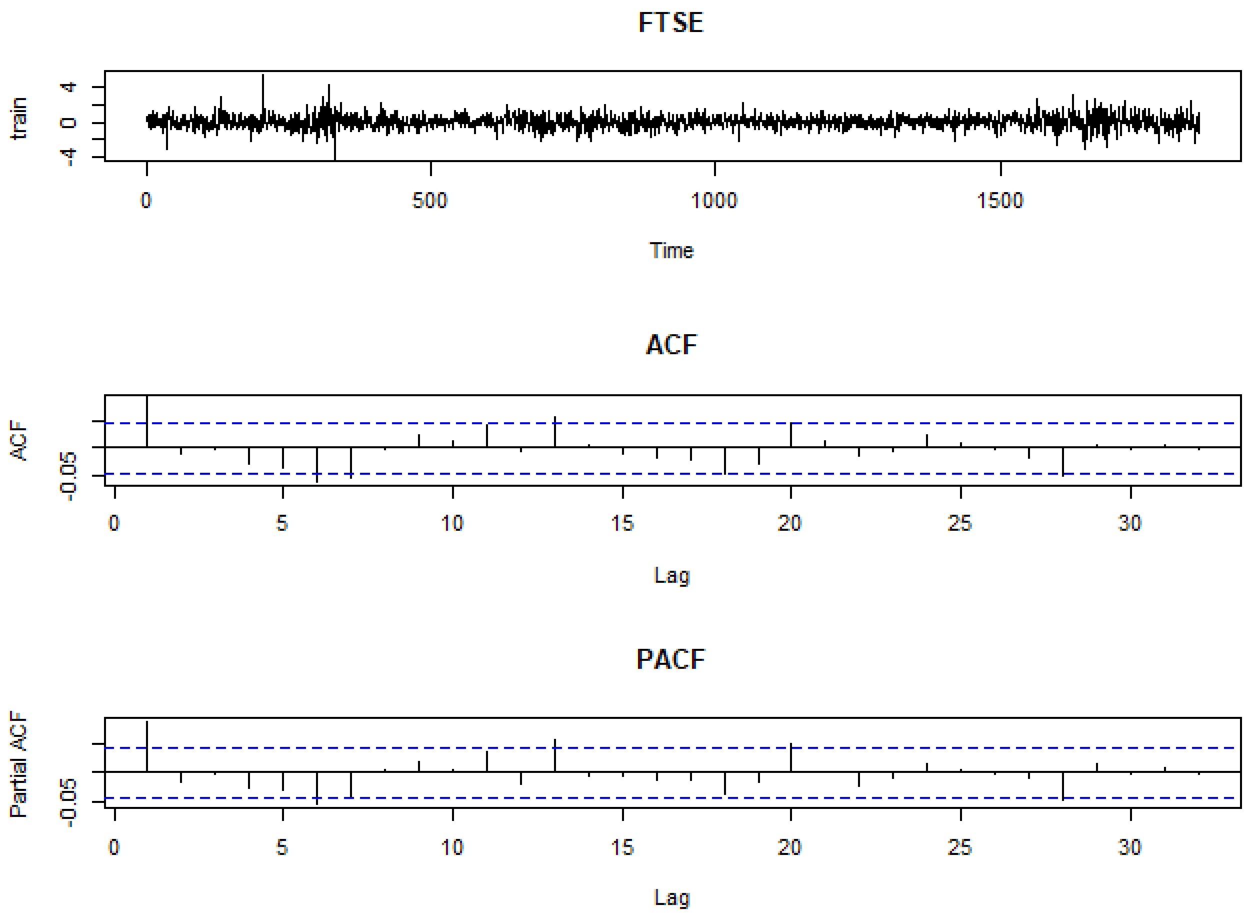

Figure 18: ACF and PACF of FTSE

Acf and pacf plots of FTSE suggest multiple posible ARMA(p,q) models (Figure 3). It can be MA(1) since acf pops up at lag 1. It can also be ARMA(1) since pacf is significant at lag 1 as well. In order to help us obtain a reasonable start, we apply function *auto.arima* to find the model with lowest “BIC” criteria.

```
> (arma <- auto.arima(train, ic='bic'))
Series: train
ARIMA(0,0,1) with zero mean

Coefficients:
         ma1
      0.0963
s.e.  0.0233

sigma^2 estimated as 0.6214:  log likelihood=-2183.21
AIC=4370.43   AICc=4370.44   BIC=4381.47
```

Accordingly, *auto.arima* suggests MA(1) to the data set. The residuals of the model, however, has heteroskedastic effect (from Ljung-Box test of squared residuals). Therefore, we should apply GARCH model to have better variance estimation.

```
> Box.test(arma$residuals^2, lag=10, type='Ljung-Box')

        Box-Ljung test

data:  arma$residuals^2
X-squared = 91.35, df = 10, p-value = 2.887e-15
```

In order to employ GARCH, we need to specify the option *mean.model* to MA(1) in function *ugarchspec* and apply this determined model into function *ugarchfit*.

```
> ug.spec <- ugarchspec(mean.model=list(armaOrder=c(0,1))) # specify garch
> (ug.fit <- ugarchfit(ug.spec, train)) # fit the model garch

...

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.050754    0.018125   2.8002 0.005107
ma1     0.084943    0.023965   3.5445 0.000393
omega   0.008631    0.004539   1.9014 0.057251
alpha1  0.042763    0.011528   3.7096 0.000208
beta1   0.943833    0.017092  55.2204 0.000000

...
```

As a result from R,

$$Var(y_t|y_{t-1}) = {\sigma_t}^2 = 0.008631 + 0.042763X_{t-1}^2 + 0.943833\sigma_{t-1}^2$$

3.2 Study conditional volatility

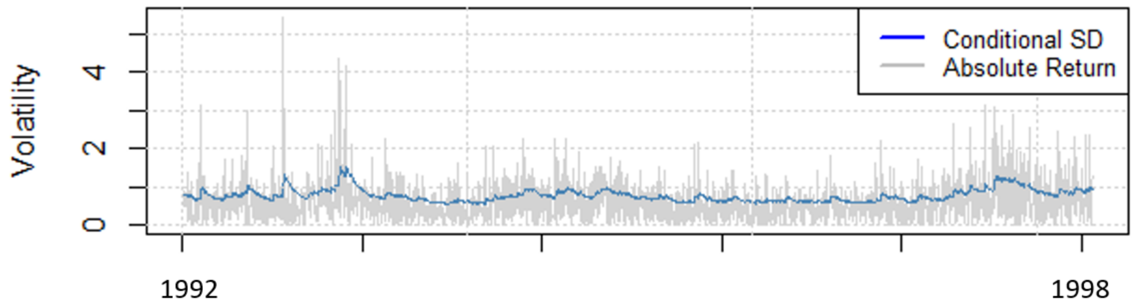

Figure 19: Conditional Standard Error of FTSE

We can plot the fitted conditional standard deviation from the model along with absolute FTSE (Figure 19). The plot demonstrates a typical GARCH type pattern: periods with large absolute FTSE return (grey line) would be followed by high conditional volatility of the next values (blue line). The exact same situation happens when we plot squared residuals and conditional variance (Figure 20).

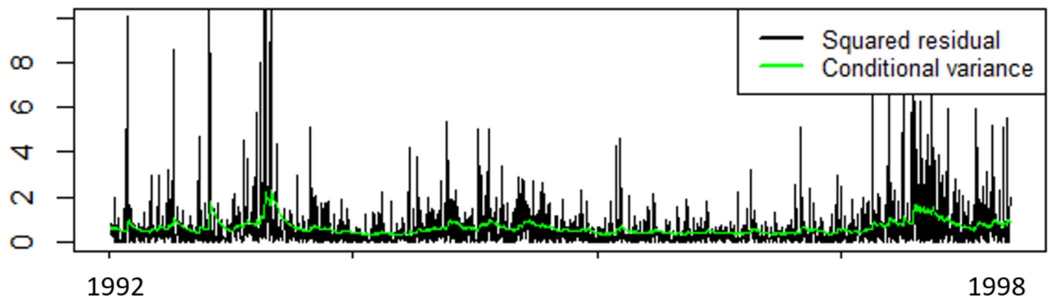

Figure 20: Conditional Variance of FTSE

3.3 Forecast

```
> (ug.fore <- ugarchforecast(ug.fit, n.ahead=10)) # forecasted values
...
       Series  Sigma
T+1  -0.06789 0.9597
T+2   0.05075 0.9578
T+3   0.05075 0.9559
T+4   0.05075 0.9540
T+5   0.05075 0.9521
T+6   0.05075 0.9503
T+7   0.05075 0.9484
T+8   0.05075 0.9466
T+9   0.05075 0.9448
T+10  0.05075 0.9431
```

By selecting option $n.ahead = 10$ in function $ugarchforecast$, we can produce the prediction of FTSE in next two weeks (10 observations in the future corresponding to 10 trading days - two weeks).

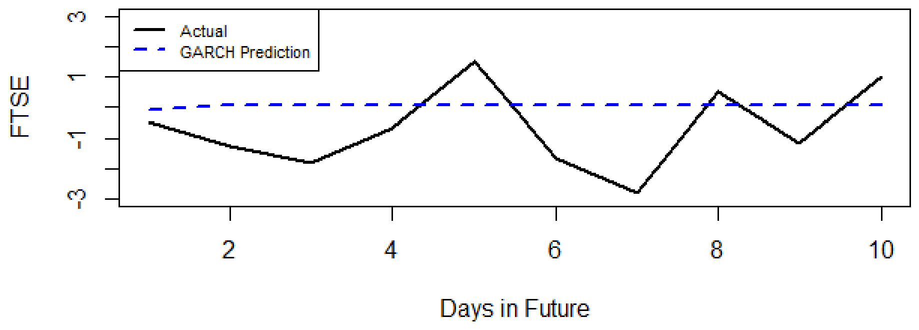

Figure 21: Forecasted Series FTSE

Figure 21 shows that except the first day, the predicted values for the next two weeks is nothing else but the average values of the data (0.05075%). Compared to the actual values (we have skipped last 10 observations in our GARCH trained model for prediction comparison), the predicted ones obviously fails to picture the daily movement of the London Stock Exchange.

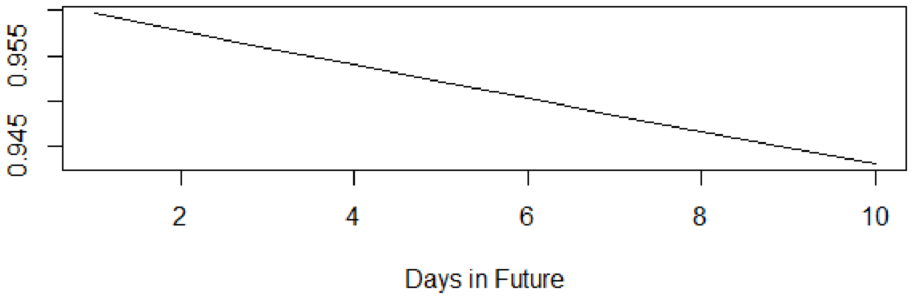

Figure 22: Forecasted Standard Deviation FTSE

Besides, the conditional volatility is expected to be decreasing (Figure 22). When we put this into the context with prior 20 observations, the forecasted volatility (expressed as forecasted variance in Figure 23) is simply the continuity of the conditional variance.

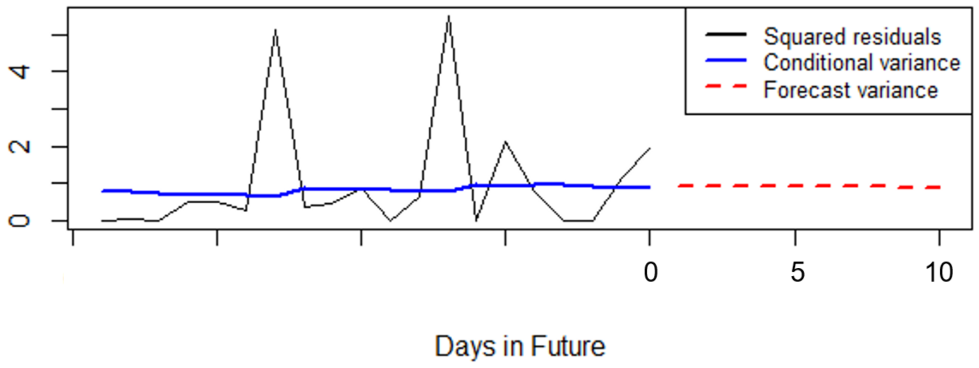

Figure 23: Forecasted Variance FTSE

In conclusion, we can not achieve a good model in this section as we did in data of SNCF. This poor result is, nonetheless, expected before the analysis. Since financial market reacts to every news including mirco-economics, macro-economics, politics or even a change in CEO position of a top blue-chip company. Therefore, employing only historic series data will be hardly adequate to make practical forecast. Moreover, if the forecast based on this simple GARCH had been able to capture the movement of stock market, then making money would have not been that difficult, wouldn't it?

References

[1] Analyst, I. (2018, September 05). What are the largest stock exchanges in the world? Retrieved from https://www.ig.com/au/trading-strategies/what-are-the-largest-stock-exchanges-in-the-world-180905/

```
# ======================================================================== #
#                             FULL CODE SCRIPT
#                       Time Series -  December 2018
#                               Khanh TRUONG
# ======================================================================== #

# This project contains analysis on two data sets:
# SNCF Traffic and UK Stock Index FTSE

### ======================================================================
### SNCF Traffic =========================================================
### ======================================================================
rm(list=ls())
library(tseries)
library(TSPred)
library(forecast)

# Prepare data ===========================================================
Y <- scan('trafic.dat')
Y <- ts(Y,start=1963,frequency=12)
X <- window(Y,end=1979+11/12)
X.test <- window(Y,start=1980)


# Fit models =============================================================
ts.plot(X, ylab='') # not stationary: there is trend and seasonality

diffX <- diff(X)
par(mfrow=c(3,1))
ts.plot(diffX, main='diffX') # there is still seasonality
abline(h=0, col='red', lty=2)
acf(diffX, main='ACF') # acf pops up at lag 12
pacf(diffX, main='PACF')

diff12X <- diff(diff(X),lag=12)
par(mfrow=c(3,1))
ts.plot(diff12X, main='diff12X') # stationary
abline(h=0, col='red', lty=2) # there is still seasonality.
acf(diff12X, main='ACF') # acf vanish after lag 13
pacf(diff12X, main='PACF') # pacf decays exponentially (less important)

# => apply MA(13) to diff12X
# => equivalently: apply SARIMA (0,1,1) (0,1,1) 12 to X
(sarima <- arima(X,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12)))


# Fit model automatically by function 'auto.arima' ========================
(sarima.auto <- auto.arima(X, trace=TRUE, ic='bic'))
# choose smallest BIC model
summary(sarima.auto)
confint(sarima.auto, level=0.95) # CI of sar1 contains 0
```

```r
t.statistic <- sarima.auto$coef['sar1']/ # estimate coefficient
  sqrt(sarima.auto$var.coef['sar1','sar1']) # standard error
pt(abs(t.statistic), df=1, lower.tail=FALSE)*2 # p-value
# insignificant different from 0, so come to the same model in sarima above


### Study residuals ==========================================================
residuals <- sarima$residuals

# residuals are stationary? --------------------------------------------------
par(mfrow=c(3,1))
ts.plot(residuals, main='Residuals')
abline(h=0, col='red',lty=2) # visually stationary

# residuals have no auto-correlation? ----------------------------------------
acf(residuals, main='ACF') # no visual correlation
pacf(residuals, main='PACF') # no visual partial correlation
Box.test(residuals, lag=12, type='Ljung-Box')
# H0: no serial correlation upto n lags
# cannot reject the null hypothesis

# volatility of residuals is stable? -----------------------------------------
Box.test(residuals^2, lag=12, type='Ljung-Box')
# H0: no arch effect
# cannot reject the null hypothesis

# distribution of residuals? -------------------------------------------------
par(mfrow=c(1,1))
hist(residuals, prob=TRUE, main=NULL, xlab=NULL, col='grey')
curve(dnorm(x, mean(residuals), sd(residuals)), lty=2, add=TRUE) # gaussian
jarque.bera.test(residuals)
# H0: residuals follow Gaussian distribution
# reject the null hypothesis


### Forecast =================================================================
(forecast <- forecast(sarima, h=12))
# will have error if import library(TSA) before,
# if so, restart R and run from the beginning.

plot(forecast)
abline(v=c(1977,1981), lty=c(2,2)) # line to zoom-in
plot(forecast, xlim=c(1977,1981), ylim=c(2500,4500)) # zoom-in

par(mfrow=c(1,1))
plotarimapred(X.test, sarima, xlim=c(1977,1981), range.percent=0.05)
legend("topleft", legend=c("Actual","Forecast"), col=c("black","blue"),
       lwd=2, cex=0.7, lty=c(2,1)) # legend
abline(v=c(1979,1981), lty=c(2,2)) # line to zoom-in

par(mfrow=c(1,1))
plotarimapred(X.test, sarima, xlim=c(1979,1981), range.percent=0.05)
legend("topleft", legend=c("Actual","Forecast"), col=c("black","blue"),
       lwd=2, cex=0.7, lty=c(2,1)) # legend
```

```
accuracy(forecast, X.test) # some accuracy metrics


### Compare to method 'decompose' ==========================================
plot(decompose(X))
# plot(decompose(X, type='multiplicative')) # method multiplicative
names(decompose(X))
random <- decompose(X)$random # residuals of decompose method

# residuals are stationary? ----------------------------------------------
par(mfrow=c(2,1))
ts.plot(random, main='Decompose')
abline(h=0, col='red',lty=2) # visually stationary
ts.plot(residuals, main='SARIMA')
abline(h=0, col='red',lty=2) # visually stationary

# residuals have no auto-correlation? ------------------------------------
random <- na.omit(random)

par(mfrow=c(2,1))
acf(random, main='Decompose') # decay exponentially (with seasonality)
acf(residuals, main='SARIMA') # no visual correlation

par(mfrow=c(2,1))
pacf(random, main='Decompose') # partial correlated upto lag 12
pacf(residuals, main='SARIMA') # no visual correlation

Box.test(random, lag=12, type='Ljung-Box')
# H0: no serial correlation upto n lags
# reject the null hypothesis

# volatility of residuals is stable? -------------------------------------
Box.test(random^2, lag=12, type='Ljung-Box')
# H0: no arch effect
# reject the null hypothesis

# distribution of residuals? ---------------------------------------------
par(mfrow=c(1,1))
hist(random, prob=TRUE, main=NULL, xlab='Residuals', col='grey',
     ylim=c(0,0.003))
curve(dnorm(x, mean(random), sd(random)), lty=2, add=TRUE) # gaussian
jarque.bera.test(random)
# H0: residuals follow Gaussian distribution
# reject the null hypothesis

# good prediction? -------------------------------------------------------
T <- time(X)
T2 <- T^2
reg <- lm(X~T+T2) # regression traffic to T and T2
summary(reg)

T.test <- data.frame(T=time(X.test), T2=time(X.test)^2) # new data
```

```r
trend.predict <- predict(reg,T.test) # predict trend based on regression
season.predict <- window(decompose(X)$seasonal,start=1979) # seasonality
X.predict <- trend.predict + season.predict # prediction based on decompose

# actual values
plot(X.test, ylim=c(2500,4500), type='l', lty=2, lwd=2, col='black',
     xlab=NULL, ylab='Traffic')
# decompose prediction
par(new=TRUE)
plot(X.predict, ylim=c(2500,4500), type='l', lty=1, lwd=2, col='red',
     ylab=NULL, xlab=NULL, xaxt='n', yaxt='n')
# SARIMA prediction
par(new=TRUE)
plot(forecast$mean, ylim=c(2500,4500), type='l', lty=1, lwd=2, col='blue',
     ylab=NULL, xlab=NULL, xaxt='n', yaxt='n')
# legend
legend("topleft",legend=c("Actual","Decompose Forecast","SARIMA Forecast"),
       col=c("black","red","blue"), lwd=2, cex=0.7, lty=c(2,1,1))
# => 'decompose' is not that bad, especially at first and last quarters



### ======================================================================
### ARCH and GARCH =======================================================
### ======================================================================
rm(list=ls())
library(TSA)
library(rugarch)

# Simulate ARCH process ==================================================
set.seed(1234)
arch <- garch.sim(n=1000,alpha = c(0.05,0.3))

# process is stationary? -------------------------------------------------
par(mfrow=c(3,1))
ts.plot(arch, main='ARCH(1)')
abline(h=0, col='red', lty=2) # visually stationary(!)

# process have no auto-correlation? --------------------------------------
acf(arch, main='ACF') # no correlation
pacf(arch, main='PACF') # no partial correlation
Box.test(arch, lag=20, type='Ljung-Box')
# H0: no serial correlation upto n lags
# cannot reject the null hypothesis

# squared series follows AR(1)? ------------------------------------------
par(mfrow=c(1,1))
pacf(arch^2, main='') # partial correlation at lag 1

# volatility of process is stable? ---------------------------------------
Box.test(arch^2, lag=20, type='Ljung-Box')
# H0: no arch effect
# reject the null hypothesis
# distribution of process? ----------------------------------------------
par(mfrow=c(1,1))
hist(arch, prob=TRUE, main=NULL, xlab='process', col='grey')
curve(dnorm(x, mean(arch), sd(arch)), lty=2, add=TRUE) # gaussian
jarque.bera.test(arch)
# H0: process follow Gaussian distribution
# reject the null hypothesis


# Simulate GARCH process ================================================
set.seed(1234)
garch <- garch.sim(n=1000, alpha = c(0.05,0.3), beta=c(0.1,0.15))

# process is stationary? ------------------------------------------------
par(mfrow=c(3,1))
ts.plot(garch, main='GARCH(1,2)')
abline(h=0, col='red', lty=2) # visually stationary(!)

# process have no auto-correlation? ---------------------------------------
acf(garch, main='ACF') # some appoach to the bounds of CI
pacf(garch, main='PACF') # some appoach to the bounds of CI
Box.test(garch, lag=20, type='Ljung-Box')
# H0: no serial correlation upto n lags
# cannot reject the null hypothesis

# squared series follows ARMA? ------------------------------------------
par(mfrow=c(2,1))
acf(garch^2, main='') # no partial correlation
pacf(garch^2, main='') # no partial correlation

# volatility of process is stable? ----------------------------------------
Box.test(garch^2, lag=20, type='Ljung-Box')
# H0: no garch effect
# reject the null hypothesis

# distribution of process? ----------------------------------------------
par(mfrow=c(1,1))
hist(garch, prob=TRUE, main=NULL, xlab='process', col='grey')
curve(dnorm(x, mean(garch), sd(garch)), lty=2, add=TRUE) # gaussian
jarque.bera.test(garch)
# H0: process follow Gaussian distribution
# reject the null hypothesis


# Validate SARIMA model =================================================
(sarima <- arima(Y,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12)))



### ====================================================================
### UK Stock Index FTSE ================================================
### ====================================================================
FTSE <- EuStockMarkets[,'FTSE']
tsp(FTSE)
```

```
FTSE.log.return <- diff(log(FTSE))*100

par(mfrow=c(2,1))
ts.plot(FTSE, ylab='', main='FTSE Index') # there is trend
ts.plot(FTSE.log.return, ylab='', main='FTSE Log Return') # no trend

# test <- window(FTSE.log.return,start=1998+0.61) # last 10 observations
# train <- window(FTSE.log.return,end=1998+0.61) # remaining

test <- tail(FTSE.log.return, 10) # last 10 observations
train <- FTSE.log.return[!(FTSE.log.return %in% test)] # remaining

par(mfrow=c(3,1))
ts.plot(train, main='FTSE')
acf(train, main='ACF') # there is still seasonality (5 lags)
pacf(train, main='PACF')

# diff.train <- diff(train) # differentiate order 1
# par(mfrow=c(3,1))
# ts.plot(diff.train, main='FTSE')
# acf(diff.train, main='ACF') # there is still seasonality (5 lags)
# pacf(diff.train, main='PACF')
#
# diff.train <- diff(train, lag=5)
# par(mfrow=c(3,1))
# ts.plot(diff.train, main='FTSE')
# acf(diff.train, main='ACF')
# pacf(diff.train, main='PACF')


# Fit models ============================================================
(arma <- auto.arima(train, ic='bic'))
Box.test(arma$residuals^2, lag=10, type='Ljung-Box')
# H0: no arch effect
# reject the null hypothesis => use GARCH model

ug.spec <- ugarchspec(mean.model=list(armaOrder=c(0,1))) # specify garch
ug.fit <- ugarchfit(ug.spec, train) # fit the model garch

par(mfrow=c(1,1))
plot(ug.fit, which=3) # wrong x-axis, should be from 1992 to 1998
legend("topright", legend=c("Conditional SD","Absolute Return"),
       col=c("blue","grey"), lwd=2, cex=0.8, lty=c(1,1))

names(ug.fit@fit)
ug.fit@fit$coef
ug.res2 <- ug.fit@fit$residuals^2 # squared residuals
ug.var <- ug.fit@fit$var # conditional variance

par(mfrow=c(1,1))
plot(ug.res2, type='l', ylab='', xlab='Days', ylim=c(0,10))
lines(ug.var, col='green')
legend("topright", legend=c("Squared residual","Conditional variance"),
       col=c("black","green"), lwd=2, cex=0.8, lty=c(1,1))
```

```r
# typical garch type pattern:
# periods with large volatility (green line),
# (squared) residuals tends to larger (black line)


# Forecast =============================================================
(ug.fore <- ugarchforecast(ug.fit, n.ahead=10)) # forecasted

# Value Forecast -------------------------------------------------------
# actual values
plot(test, type='l', lty=1, lwd=2, col='black',
     xlab=NULL, ylab=NULL, ylim=c(-3,3), xaxt='n', yaxt='n')
# GARCH prediction
par(new=TRUE)
plot(ug.fore@forecast$seriesFor, type='l', lty=2, lwd=2, col='blue',
     ylab='FTSE', xlab='Days in Future', ylim=c(-3,3))
# legend
legend("topleft",legend=c("Actual","GARCH Prediction"),
       col=c("black","blue"), lwd=2, cex=0.7, lty=c(1,2))

# Volatility Forecast --------------------------------------------------
ug.f <- ug.fore@forecast$sigmaFor # standard error of forecasted values
plot(ug.f, type = "l", ylab='',
     xlab='Days in Future') # standard error decreasing

# gets the last 20 observations of squared residuals
ug.res2.t <- c(tail(ug.res2,20),rep(NA,10))
# gets the last 20 observations of conditional variance
ug.var.t <- c(tail(ug.var,20),rep(NA,10))
# variance of forecasted values
ug.f <- c(rep(NA,20),(ug.f)^2)

plot(ug.res2.t, type="l", xlab='Days in Future', ylab='')
lines(ug.var.t, col="blue", lwd=2)
lines(ug.f, col="red", lty=2, lwd=2)
legend("topright", legend=c("Squared residuals","Conditional variance",
  "Forecast variance"), col=c("black","blue","red"), lwd=2, cex=0.8,
  lty=c(1,1,2))

# ====================================================================== #
```