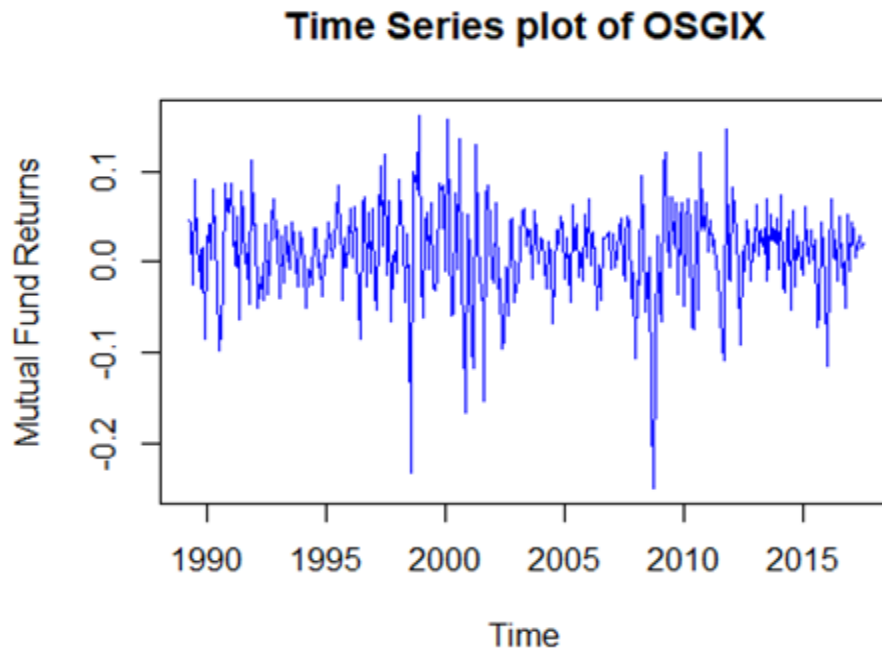


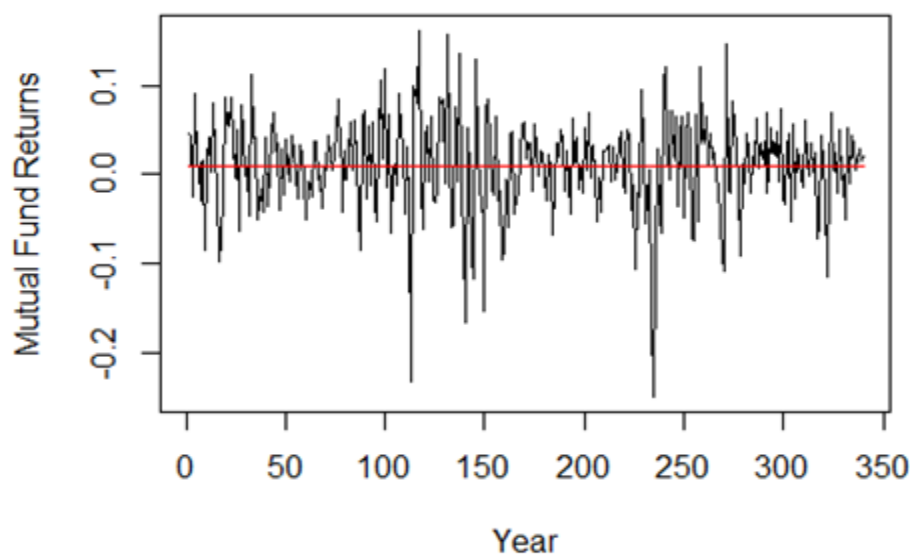
REPORT IN OSGIX FUND

1) Αρχικά αναπαριστώντας το fund OSGIX σαν χρονοσειρά μπορούμε να κάνουμε τις εξής παρατηρήσεις:

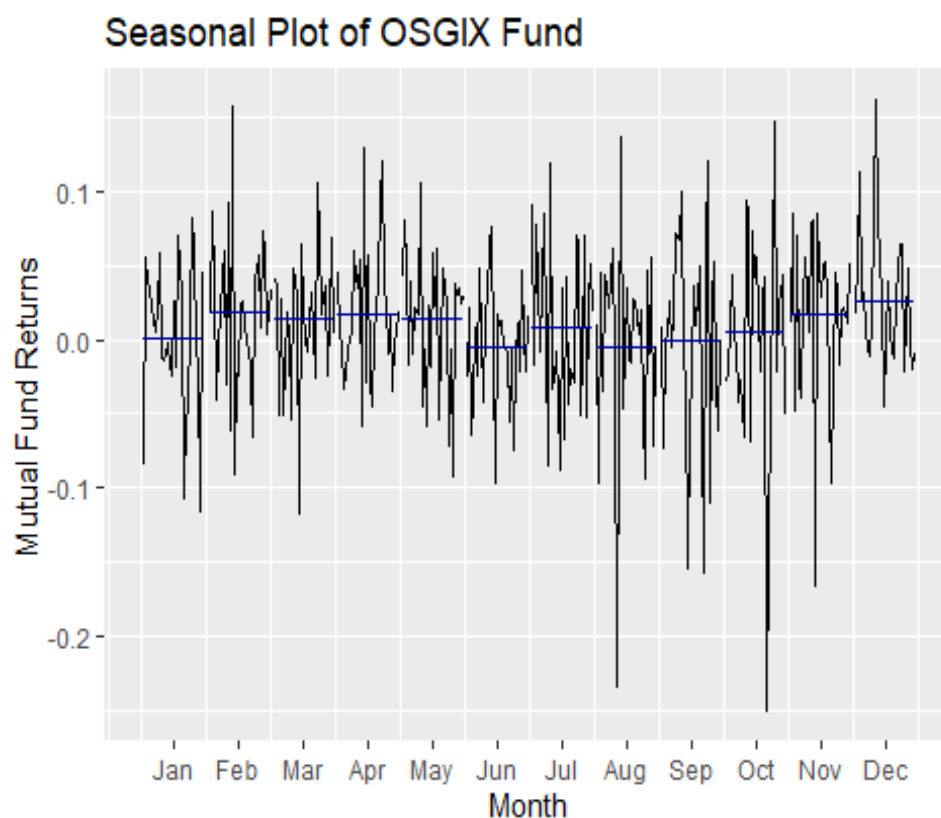
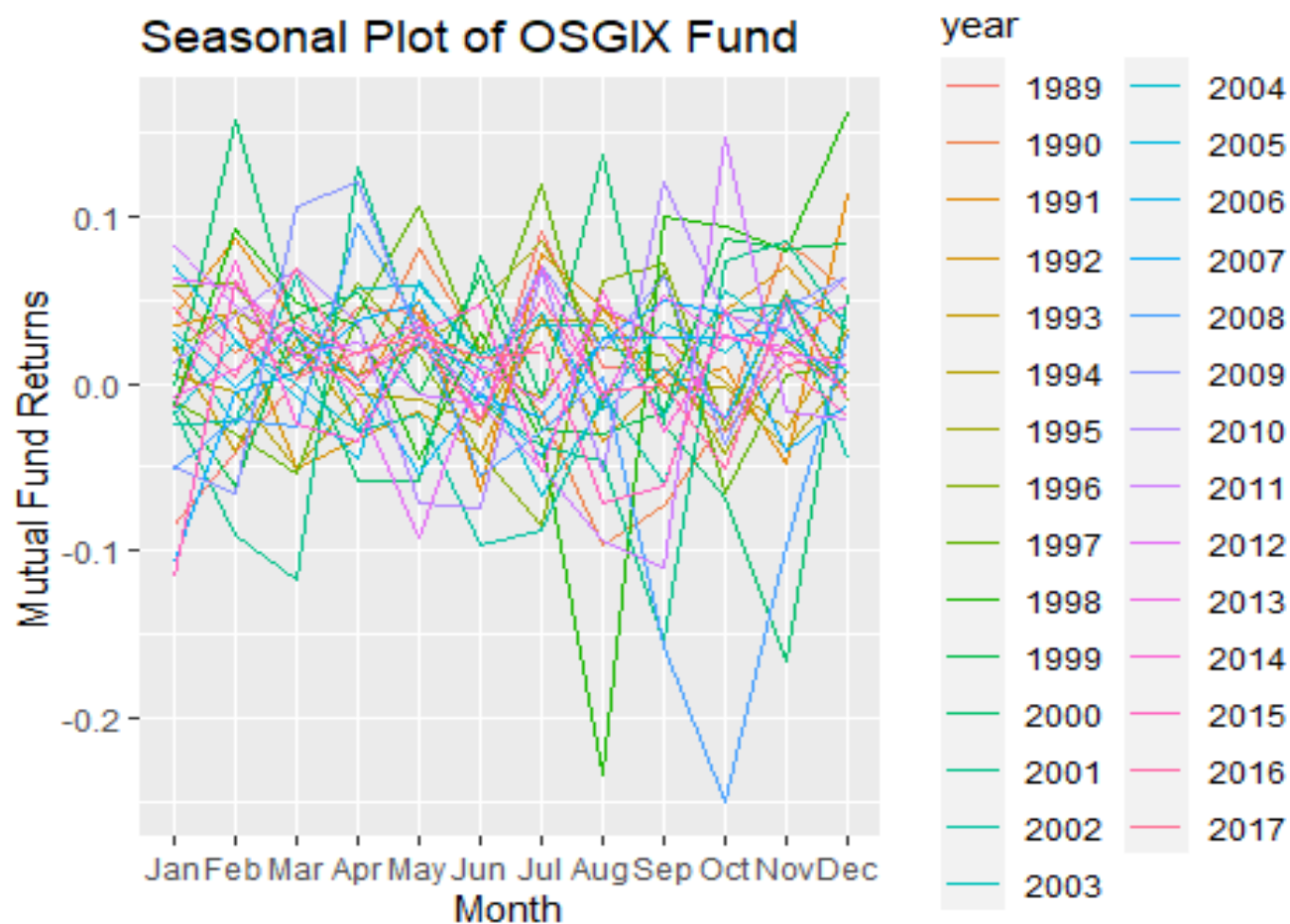


- Ενδείξεις για στάσιμη σειρά η οποία κινείται γύρω από το 0, χωρίς να υπάρχει τάση
- Παρατηρούμε volatility clustering, άρα αναμένουμε να έχουμε ετεροσκεδαστικότητα, λόγω της έντονης μεταβλητότητας (άρα θα χρειαστούμε Garch μοντέλα)
- Δεν φαίνεται να υπάρχουν δομικές αλλαγές στην σειρά

Change points for OSGIX period 1989-2017

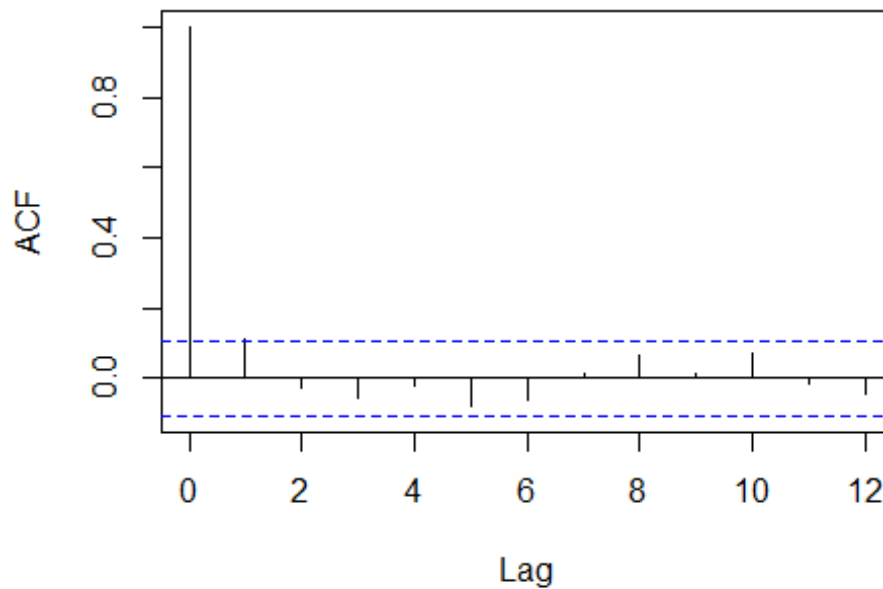


Η επιβεβαίωση, ότι δεν υπάρχουν δομικές αλλαγές στην σειρά μας , καθώς και ότι κινείται γύρω από το μηδέν. Ας δούμε οπτικά εάν υπάρχει εποχικότητα ως εξής :



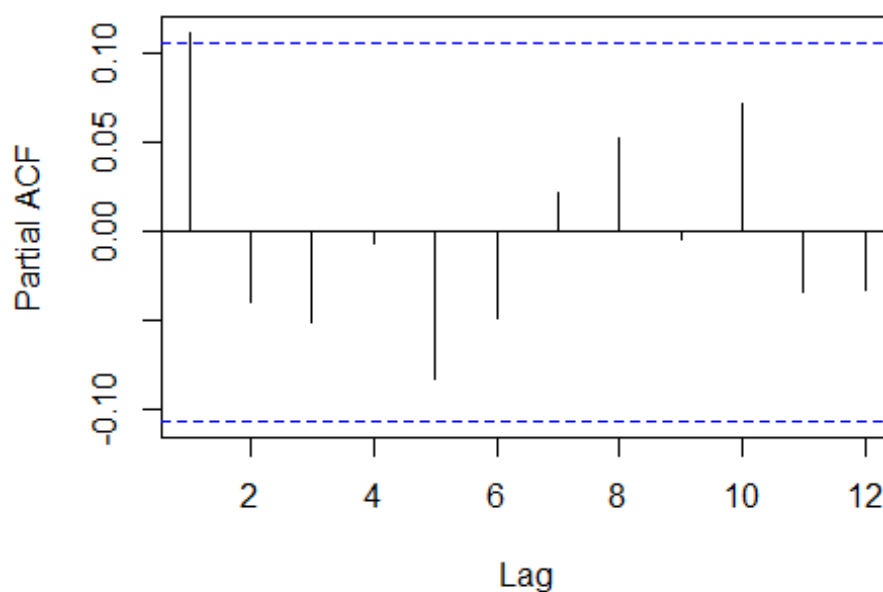
Παρατηρούμε ότι δεν υπάρχουν μεγάλες αποκλίσεις μεταξύ των μέσων των διαφορετικών μηνών άρα η σειρά δεν έχει εποχικότητα. Ας δούμε αν θα χρειαστούμε ARMA(p,q) όρους για να λύσουμε ενδεχόμενο πρόβλημα αυτοσυσχέτισης των αποδόσεων της χρονοσειράς OSGIX.

ACF plot of OSGIX



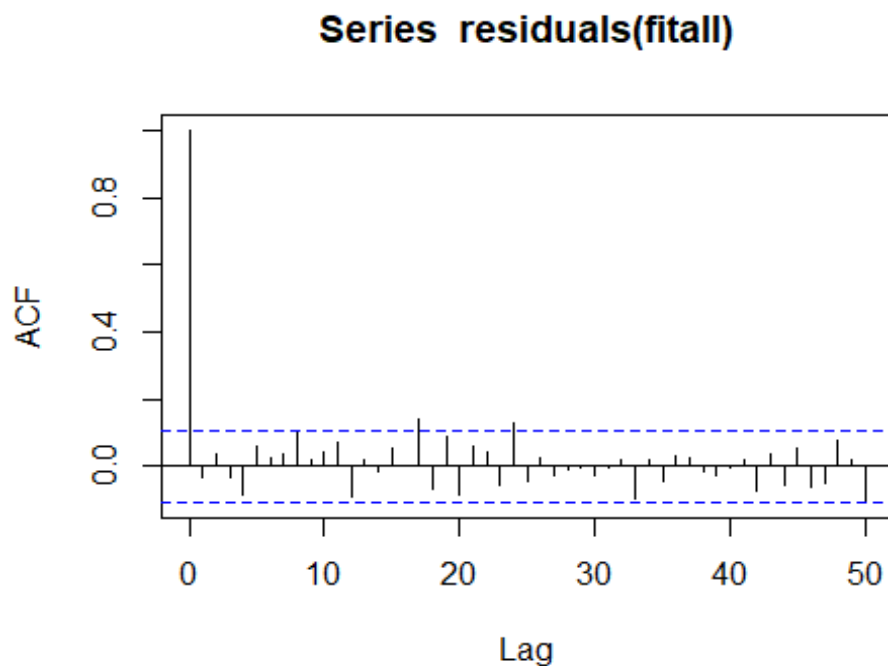
Εφόσον όλα τα lags μέχρι το 12 δεν είναι στατιστικά σημαντικά (με εξαίρεση το πρώτο που είναι για λίγο εκτός των ορίων, δηλαδή στατιστικά σημαντικό) έχουμε ενδείξεις ότι δεν υπάρχει πρόβλημα αυτοσυσχέτισης (καθώς και για την στασιμότητα της χρονοσειράς). Ας δούμε και το διάγραμμα μερικής αυτοσυσχέτισης για το fund OSGIX .

PACF plot of OSGIX



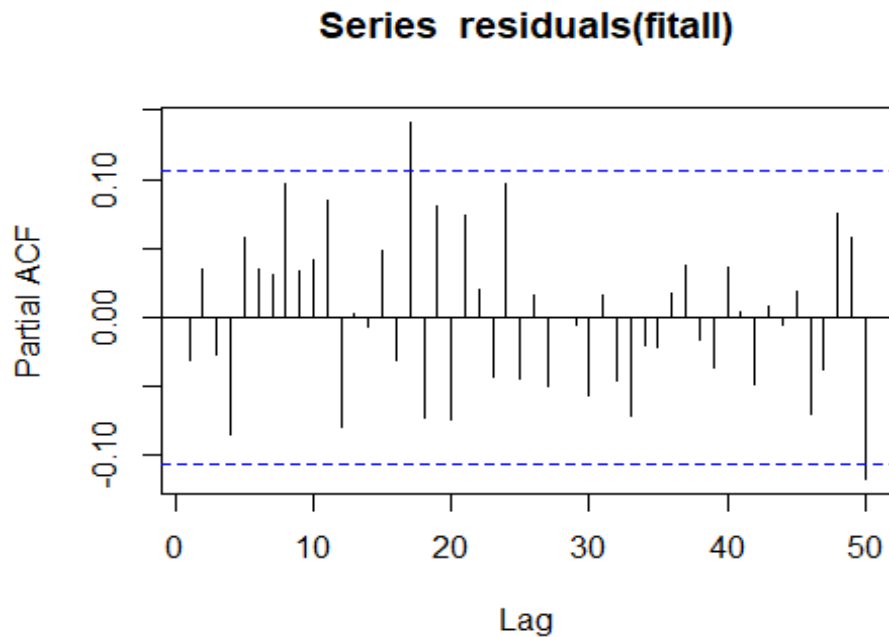
Ενισχύονται οι ενδείξεις, ότι δεν υπάρχει πρόβλημα αυτοσυσχέτισης για το fund OSGIX. Η επιβεβαίωση έρχεται μέσω του ελέγχου Box-Pierce (ή Ljung-Box), όπου τα αποτελέσματα οδηγούν στο να μην απορρίπτουμε την μηδενική υπόθεση άρα δεν υπάρχει αυτοσυσχέτιση. Συνεπώς το πρώτο lag που ήταν για λίγο εκτός των ορίων ήταν λευκός θόρυβος.

2) Εξερευνούμε τις γραμμικές σχέσεις μεταξύ των παραγόντων που επηρεάζουν το fund της OSGIX. Στην συνέχεια, εφαρμόζοντας την πολλαπλή γραμμική παλινδρόμηση των έξι παραγόντων με τις αποδόσεις του fund OSGIX παίρνουμε συντελεστή προσδιορισμού 89% και προσαρμοσμένο συντελεστή προσδιορισμού σχεδόν 89%, γεγονός που σημαίνει ότι έχουμε καλή προσαρμογή του μοντέλου. Με χρήση των μεθόδων forward, backward και stepwise βλέπουμε ότι καταλήγουν στο ίδιο συμπέρασμα προκρίνοντας σαν βέλτιστο μοντέλο, αυτό που δεν περιέχει την Χ4 επεξηγηματική μεταβλητή. Στην συνέχεια για να μελετήσουμε τα κατάλοιπα του μοντέλου της πολλαπλής γραμμικής παλινδρόμησης σχεδιάζουμε τα autocorrelation και partial autocorrelation plots των καταλοίπων :



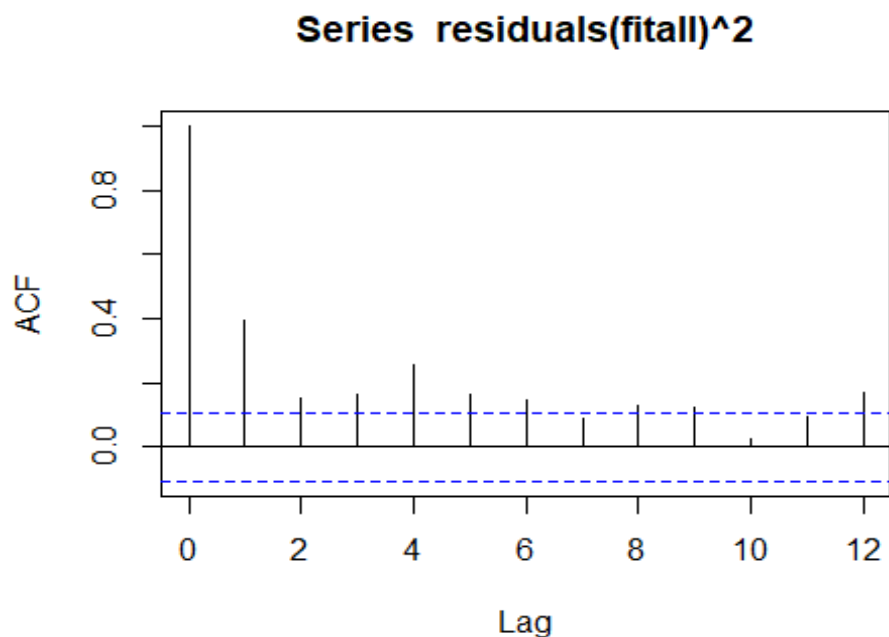
Στο διάγραμμα αυτοσυσχέτισης των καταλοίπων παρατηρούμε ότι 2 στα 50 lags βρίσκονται εκτός των ορίων,είναι δηλαδή στατιστικά σημαντικά,συνεπώς έχουμε ενδείξεις ότι τα κατάλοιπα δεν έχουν

αυτοσυσχέτιση.

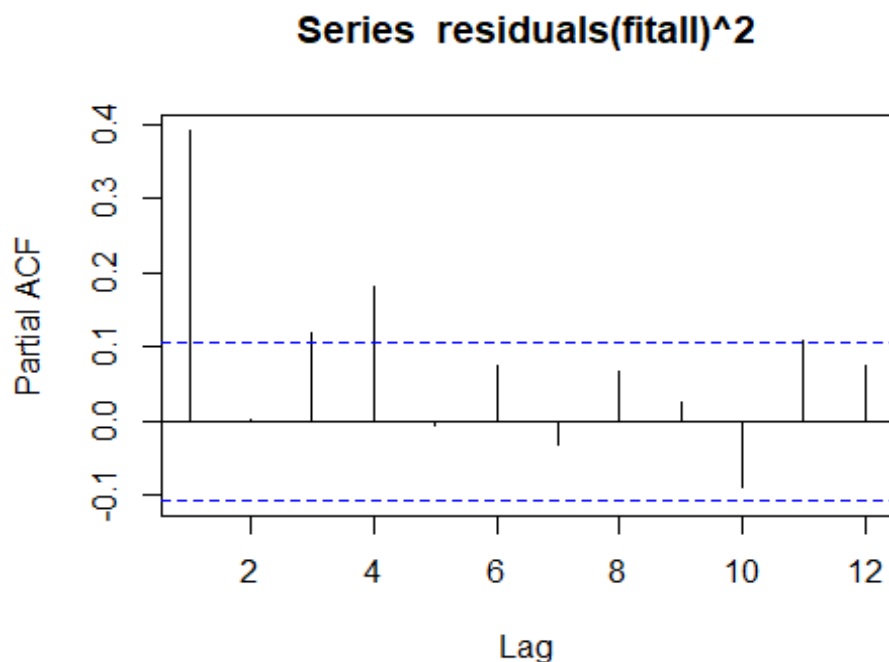


Από το διάγραμμα της μερικής αυτοσυσχέτισης των καταλοίπων παρατηρούμε 1 στα 50 lag εκτός των ορίων οπότε ενισχύονται οι ενδείξεις που έχουμε ότι δεν υπάρχει πρόβλημα αυτοσυσχέτισης στα κατάλοιπα της παλινδρόμησης. Πραγματοποιώντας τόσο το Box-Pierce test όσο και το Ljung-Box test τα αποτελέσματα συνάδουν και μας δίνουν την επιβεβαίωση, ότι δεν υφίσταται αυτοσυσχέτιση των καταλοίπων. Έτσι, τα lags που βρίσκονταν εκτός των διακεκομμένων γραμμών ήταν White Noise. Σαν άμεση συνέπεια, δεν χρειάζεται να χρησιμοποιήσουμε ARMA όρους (ώστε να διορθώσουμε το ενδεχόμενο πρόβλημα αυτοσυσχέτισης των καταλοίπων).

Ας δούμε τα διαγράμματα αυτοσυσχέτισης και μερικής αυτοσυσχέτισης των τετραγώνων των καταλοίπων της πολλαπλής παλινδρόμησης :



Παρατηρούμε πως υπάρχει πρόβλημα ετεροσκεδαστικότητας στα lags 1,2,3,4,5,6,8,9,12, άρα θα χρειαστούμε Garch όρους ώστε να διορθώσουμε το πρόβλημα. Ας δούμε και το διάγραμμα των τετραγώνων των καταλοίπων της παλινδρόμησης :

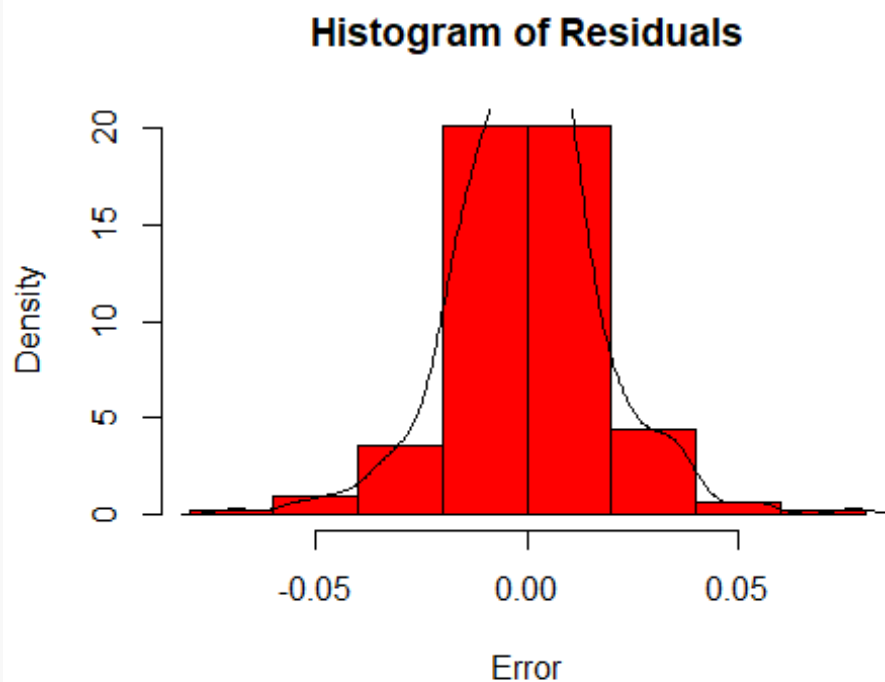


Παρατηρούμε ότι υπάρχει πρόβλημα ετεροσκεδαστικότητας στα lags 1,3,4,11, συνεπώς θα χρειαστούμε ARCH όρους, ώστε να το διορθώσουμε. Δοκιμάζοντας το Garch(1,1) μοντέλο παρατηρούμε ότι :

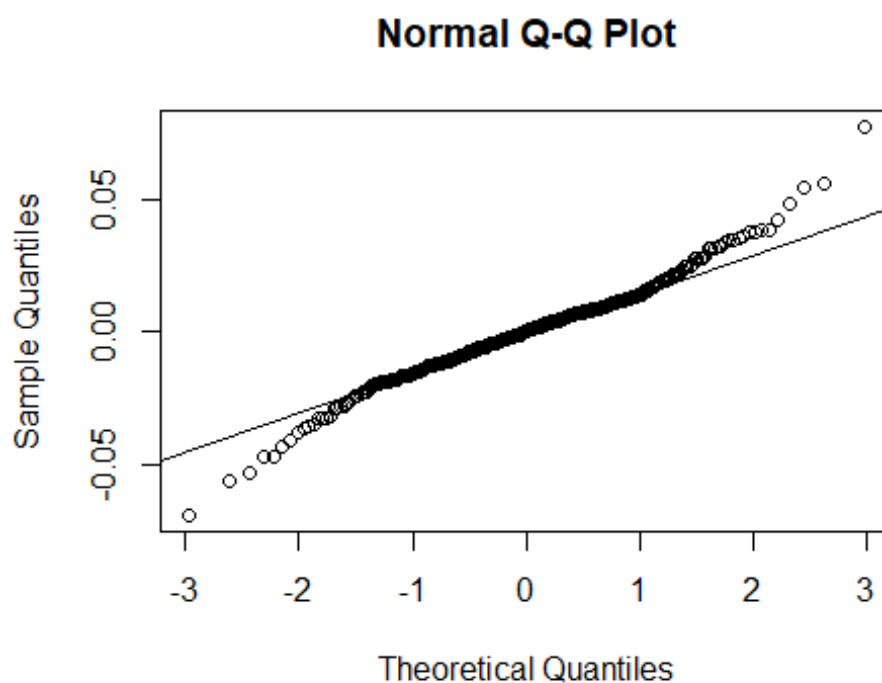
```
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = y, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x000000002a8c3250>
##  [data = y]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##           mu      omega    alpha1    beta1
## 0.01005907 0.00014035 0.14198218 0.81643243
##
## Std. Errors:
## based on Hessian
## Error Analysis:
```

```
##           Estimate Std. Error t value Pr(>|t|)
## mu       1.006e-02  2.564e-03   3.922 8.76e-05 ***
## omega    1.403e-04  8.095e-05   1.734 0.08297 .
## alpha1   1.420e-01  4.582e-02   3.099 0.00194 **
## beta1    8.164e-01  5.531e-02  14.760 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Log Likelihood:
## 529.4718      normalized: 1.55727
## Description:
## Tue Apr 27 17:28:30 2021 by user: mihal
## Standardised Residuals Tests:
##                                     Statistic p-Value
## Jarque-Bera Test      R      Chi^2  84.00925  0
## Shapiro-Wilk Test     R      W      0.9692981 1.316587e-06
## Ljung-Box Test        R      Q(10) 13.14746 0.2155437
## Ljung-Box Test        R      Q(15) 17.59582 0.28451
## Ljung-Box Test        R      Q(20) 20.75612 0.41161
## Ljung-Box Test        R^2  Q(10)  3.729235 0.9587361
## Ljung-Box Test        R^2  Q(15)  5.934771 0.9808227
## Ljung-Box Test        R^2  Q(20)  7.575184 0.994326
## LM Arch Test          R      TR^2   4.237143 0.978767
##
## Information Criterion Statistics:
##           AIC           BIC           SIC           HQIC
## -3.091011 -3.045964 -3.091283 -3.073062
```

Συνεπώς, είναι ορατό πως υπάρχει πρόβλημα στην κανονικότητα (ελέγχοντας τα p_value των ελέγχων Jarque-Bera και Shapiro-Wilk βλέπουμε ότι απορρίπτεται η μηδενική υπόθεση που υποστηρίζει την ύπαρξη κανονικότητας). Επιπλέον, μπορούμε να πάρουμε μια ιδέα για το πως είναι κατανομημένα τα κατάλοιπα με χρήση του ιστογράμματος :



Όπως είναι εύκολα ορατό αναμένουμε να έχουμε πρόβλημα στην κανονικότητα των καταλοίπων. Επιπρόσθετα, μπορούμε να δούμε που ακριβώς υπάρχει το πρόβλημα της μη κανονικότητας των καταλοίπων με χρήση του QQ-PLOT :



Είναι προφανές λοιπόν, ότι το πρόβλημα της μη κανονικότητας των καταλοίπων εντοπίζεται στις ουρές!

Ακόμα, βλέπουμε ότι μέχρι και το lag20 δεν υπάρχει πρόβλημα αυτοσυσχέτισης των καταλοίπων, ενώ διορθώθηκε και το πρόβλημα της ετεροσκεδαστικότητας (ορατό τόσο μέσω των Ljung-Box test όσο και από το LM Arch test).

3) Στην συνέχεια υπολογίζουμε το βέλτιστο Garch(p,q) μοντέλο βάσει των πληροφοριακών κριτηρίων AIC,BIC. Γνωρίζουμε ότι η επιλογή των ARCH όρων θα γίνει βάσει του διαγράμματος μερικής αυτοσυσχέτισης των τετραγώνων των καταλοίπων (ελέγχοντας ποια lags είναι στατιστικά σημαντικά), ενώ η επιλογή των Garch όρων θα γίνει βάσει του διαγράμματος αυτοσυσχέτισης των τετραγώνων των καταλοίπων (ελέγχοντας ποια lags είναι στατιστικά σημαντικά). Συγκρίνοντας τα AIC των Garch(p,q) μοντέλων καταλήγουμε ότι βέλτιστο μοντέλο βάσει AIC είναι το Garch(1,2), καθώς παρουσιάζει την μικρότερη τιμή AIC. Κάνοντας τις συγκρίσεις των Garch(p,q) μοντέλων βάση του BIC, καταλήγουμε πως καλύτερο μοντέλο είναι το Garch(1,1), καθώς παρουσιάζει το μικρότερο BIC. Αφού καταλήξαμε στο βέλτιστο Garch μοντέλο , τρέχουμε την πολλαπλή παλινδρόμηση με το Garch(1,1) μοντέλο. Πραγματοποιούμε backward elimination method χειροκίνητα με σκοπό να καταλήξουμε στο βέλτιστο μοντέλο, το οποίο περιλαμβάνει όλες τις στατιστικά σημαντικές επεξηγηματικές μεταβλητές. Καταλήγουμε ότι το μοντέλο αυτό, είναι το εξής :

```
##
## *-----*
## *          GARCH Model Fit          *
## *-----*
##
## Conditional Variance Dynamics
## -----
## GARCH Model   : sGARCH(1,1)
## Mean Model    : ARFIMA(0,0,0)
## Distribution   : norm
##
## Optimal Parameters
## -----
##          Estimate Std. Error t value Pr(>|t|)
## mu      0.000911   0.000804   1.1336 0.256962
## mxreg1   0.010553   0.000210  50.1520 0.000000
## mxreg2   0.003895   0.000312  12.4643 0.000000
## mxreg3  -0.003966   0.000464  -8.5466 0.000000
## mxreg4   0.001149   0.000194   5.9293 0.000000
## omega    0.000035   0.000015   2.3513 0.018710
## alpha1   0.276106   0.080126   3.4459 0.000569
## beta1    0.615216   0.103090   5.9678 0.000000
##
## Robust Standard Errors:
##          Estimate Std. Error t value Pr(>|t|)
## mu      0.000911   0.000820   1.1112 0.266468
## mxreg1   0.010553   0.000226  46.6569 0.000000
## mxreg2   0.003895   0.000356  10.9296 0.000000
## mxreg3  -0.003966   0.000491  -8.0715 0.000000
## mxreg4   0.001149   0.000203   5.6651 0.000000
## omega    0.000035   0.000015   2.3944 0.016649
## alpha1   0.276106   0.076915   3.5898 0.000331
## beta1    0.615216   0.098265   6.2608 0.000000
##
## LogLikelihood : 924.8277
```

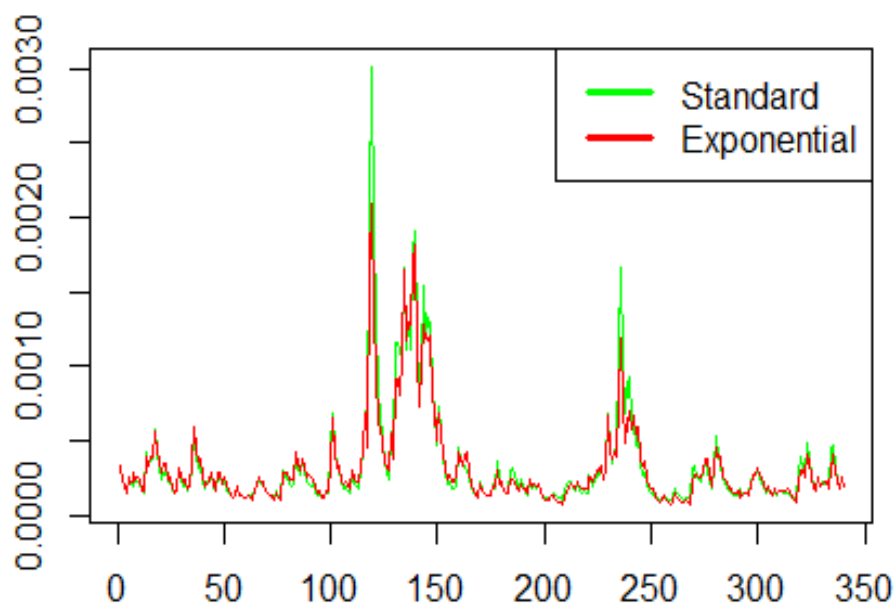
Όπως είναι εμφανές από τα p_value , οι 4 επεξηγηματικές μεταβλητές που έμειναν, είναι όλες στατιστικά σημαντικές. Θα διορθώσουμε λοιπόν το πρόβλημα της μη κανονικότητας χρησιμοποιώντας μια κατανομή που συλλαμβάνει καλύτερα τις παχιές ουρές όπως την Student-t κατανομή (ή την γενικευμένη κανονική κατανομή). Αυτό μπορούμε να το διαπιστώσουμε εύκολα μέσω του qq-plot των standardized residuals. Έτσι, καταλήγουμε στο βέλτιστο μοντέλο το οποίο περνάει τον διαγνωστικό έλεγχο (ο οποίος αποτελεί το τρίτο βήμα της Box-Jenkins μεθοδολογίας), άρα είναι αξιόπιστο προκειμένου να το χρησιμοποιήσουμε ώστε να κάνουμε πρόβλεψη. Όσο αφορά την ερμηνεία του μοντέλου, έχουμε ότι το εκτιμώμενο μ (η σταθερά του υποδείγματος) είναι σχεδόν 0.0001. Δηλαδή, η μέση απόδοση του fund OSGIX είναι περίπου 1% αλλά δεν είναι στατιστικά σημαντική σε επίπεδο σημαντικότητας $\alpha=5\%$ καθώς έχει p_value 0.25. Εάν μας ενδιαφέρει να συλλάβουμε το leverage effect, το οποίο δεν συλλαμβάνουν τα Garch μοντέλα θα μπορούσαμε να χρησιμοποιήσουμε Egarch μοντέλα. Στην προκειμένη, το Egarch(1,1) χρησιμοποιώντας student-t κατανομή σε συνδυασμό με όλους τους regressors θα ήταν το βέλτιστο μοντέλο βάση AIC. Εάν θέλουμε να δούμε τις διαφορές στην μεταβλητότητα του standard garch με το exponential garch μπορούμε να ελέγξουμε τον βαθμό συσχέτισης τους :

```
cor(voly_Sgarch, voly_Egarch)
```

```
## [1] 0.9769809
```

Επίσης, μπορούμε να τις δούμε μαζί σε ένα διάγραμμα ως εξής :

Time Series Plot of Volatily of SGarch vs EGarch



Παρατηρούμε ότι το standard garch έχει μεγαλύτερη μεταβλητότητα σε κάποια σημεία, σε σχέση με το exponential garch. Η μεταβλητότητα αποτελεί ένα μέτρο του κινδύνου, συνεπώς μεγαλύτερη μεταβλητότητα οδηγεί σε μεγαλύτερο κίνδυνο.

4) Πρώτος τρόπος για να κάνουμε πρόβλεψη, είναι μέσω της εντολής `ugarchforecast`, ως εξής :

```
##
## *-----*
## *          GARCH Model Forecast          *
## *-----*
## Model: sGARCH
## Horizon: 24
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=1970-12-07 02:00:00]:
##      Series   Sigma
## T+1  0.00103  0.01250
## T+2  0.00103  0.01324
## T+3  0.00103  0.01386
## T+4  0.00103  0.01438
## T+5  0.00103  0.01484
## T+6  0.00103  0.01523
## T+7  0.00103  0.01557
## T+8  0.00103  0.01587
## T+9  0.00103  0.01613
## T+10 0.00103  0.01636
## T+11 0.00103  0.01656
## T+12 0.00103  0.01673
## T+13 0.00103  0.01689
## T+14 0.00103  0.01703
## T+15 0.00103  0.01715
## T+16 0.00103  0.01725
## T+17 0.00103  0.01735
## T+18 0.00103  0.01743
## T+19 0.00103  0.01751
## T+20 0.00103  0.01757
## T+21 0.00103  0.01763
## T+22 0.00103  0.01768
## T+23 0.00103  0.01773
## T+24 0.00103  0.01777
```

Το παραπάνω μας δίνει τις τιμές των Y_t (σημειακή εκτίμηση) στην αριστερή στήλη (series), ενώ στην δεξιά στήλη έχουμε τις τιμές των δεσμευμένων διακυμάνσεων (sigma).

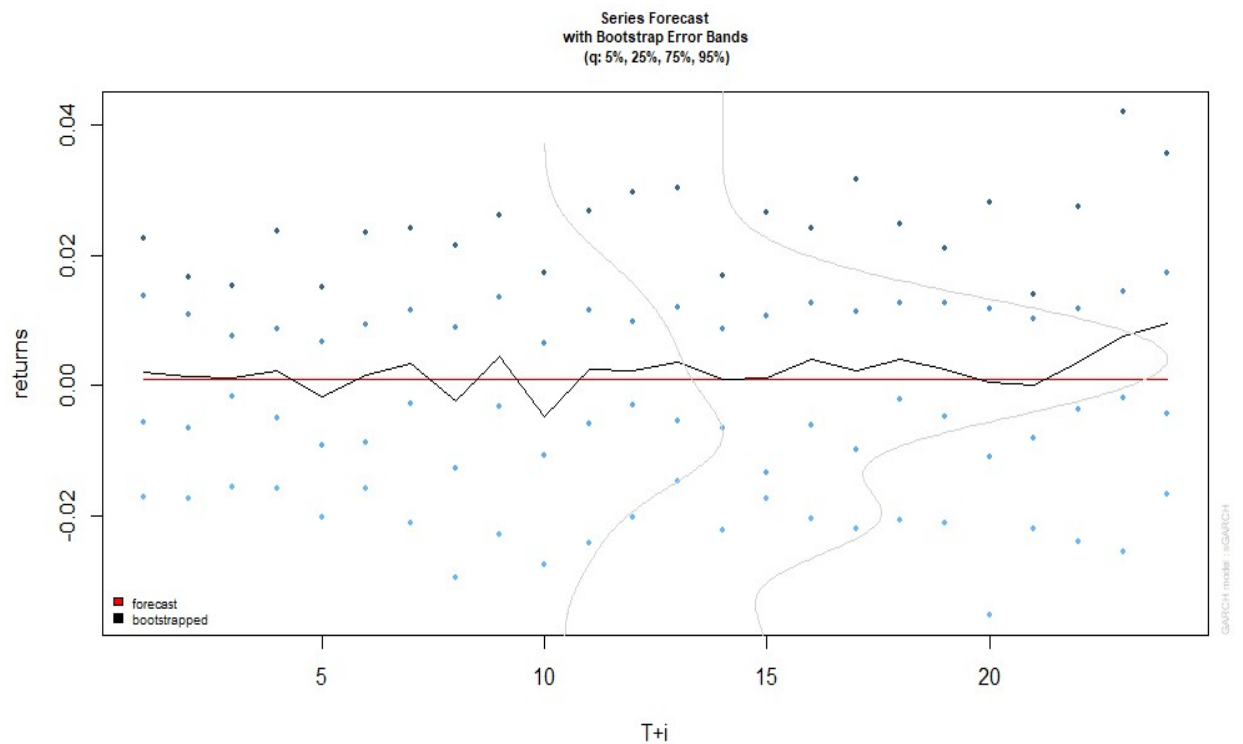
- Εναλλακτικός τρόπος πρόβλεψης είναι μέσω του Garch Bootstrap :

```
bootp <- ugarchboot(modelres4, method = c("Partial", "Full")[1], n.ahead = 24, n.bootpred = 24)
bootp
##
## *-----*
## *          GARCH Bootstrap Forecast          *
## *-----*
```

```

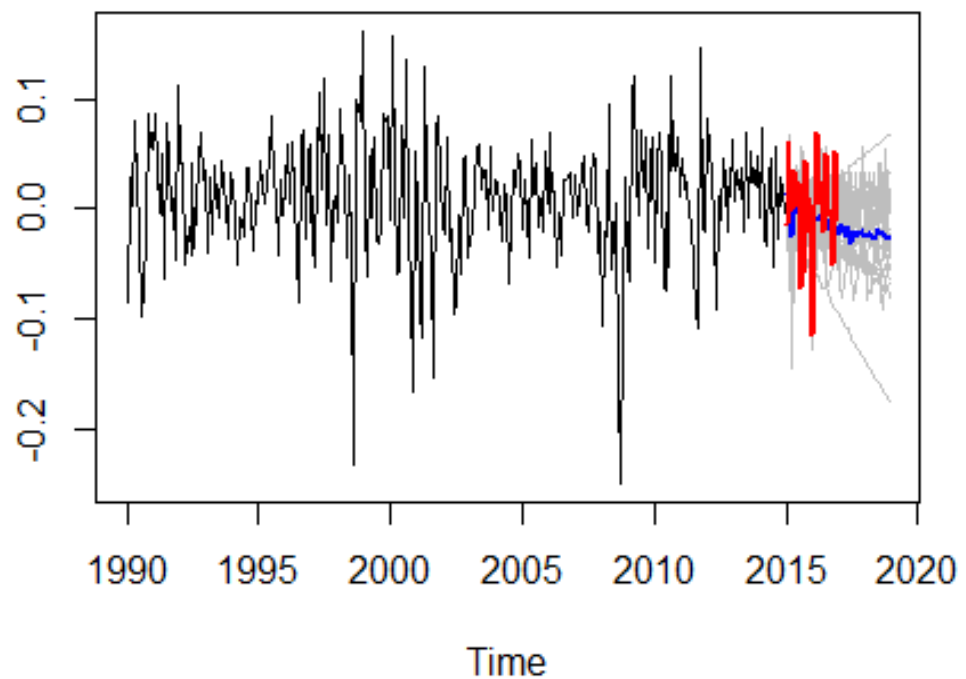
## Model : sGARCH
## n.ahead : 24
## Bootstrap method: partial
## Date (T[0]): 1970-12-07 02:00:00
##
## Series (summary):
##          min          q.25          mean          q.75          max forecast[ana
lytic]
## t+1 -0.017154 -0.001252  0.002354 0.009030 0.012386          0
.00103
## t+2 -0.021332 -0.008064  0.002507 0.009562 0.024638          0
.00103
## t+3 -0.021990 -0.012958 -0.002289 0.004855 0.054968          0
.00103
## t+4 -0.056254 -0.010721 -0.000064 0.013043 0.032601          0
.00103
## t+5 -0.036740 -0.006103 -0.002205 0.005253 0.011194          0
.00103
## t+6 -0.025821 -0.015319 -0.005630 0.001359 0.018813          0
.00103
## t+7 -0.019915 -0.006846  0.002019 0.008541 0.022803          0
.00103
## t+8 -0.046613 -0.001332  0.002689 0.013316 0.025160          0
.00103
## t+9 -0.046189 -0.009957 -0.000503 0.011466 0.026719          0
.00103
## t+10 -0.041015  0.001453  0.009403 0.020464 0.042190          0
.00103
## .....
##
## Sigma (summary):
##          min          q0.25          mean          q0.75          max forecast[analyt
ic]
## t+1  0.012503 0.012503 0.012503 0.012503 0.012503          0.012
503
## t+2  0.011476 0.011523 0.012113 0.012472 0.014958          0.013
236
## t+3  0.010869 0.011729 0.013055 0.013879 0.016576          0.013
855
## t+4  0.011301 0.011902 0.014155 0.014436 0.031407          0.014
384
## t+5  0.010978 0.012430 0.015336 0.015695 0.039405          0.014
839
## t+6  0.010507 0.011860 0.014543 0.014122 0.033293          0.015
232
## t+7  0.010178 0.012479 0.014795 0.015663 0.027971          0.015
573
## t+8  0.010002 0.012210 0.014067 0.014382 0.024077          0.015
871
## t+9  0.010963 0.012067 0.014955 0.016660 0.031973          0.016
131
## t+10 0.011224 0.011761 0.015374 0.016861 0.035802          0.016
359

```



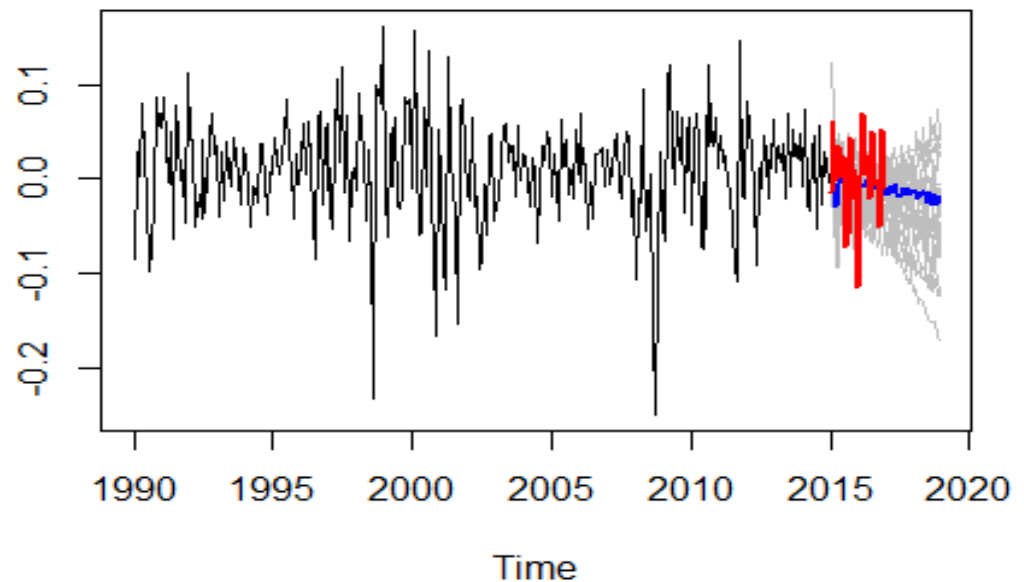
- Ακόμα, μπορούμε να δούμε την πρόβλεψη με χρήση νευρωνικών δικτύων και συγκεκριμένα μέσω multiple layer perceptron (default mlp model has 5 hidden nodes) :

Forecasts from MLP



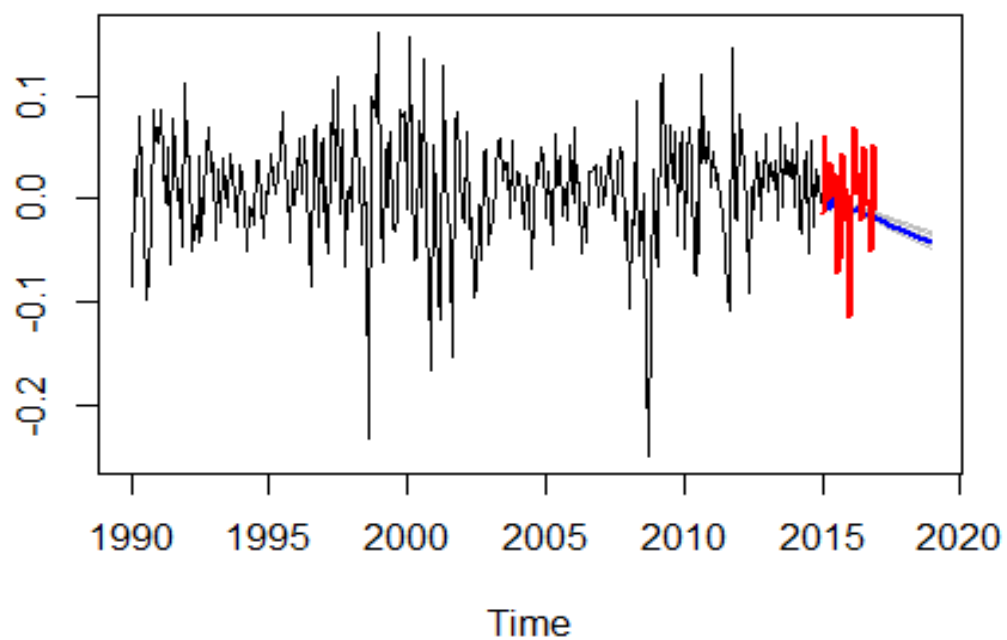
- Ένας δεύτερος τρόπος να δούμε διαγραμματικά την πρόβλεψη με χρήση νευρωνικών δικτύων είναι με ένα mlp model with the number of hidden nodes determined by 20% validation : (διαδικασία που διήρκεσε 13 λεπτά στον υπολογιστή μου)

Forecasts from MLP



- Ένας τρίτος τρόπος να δούμε διαγραμματικά την πρόβλεψη με χρήση νευρωνικών δικτύων είναι με ένα mlp model with the number of hidden nodes determined by 5 fold cross validation : (διαδικασία που διήρκεσε περισσότερο)

Forecasts from MLP

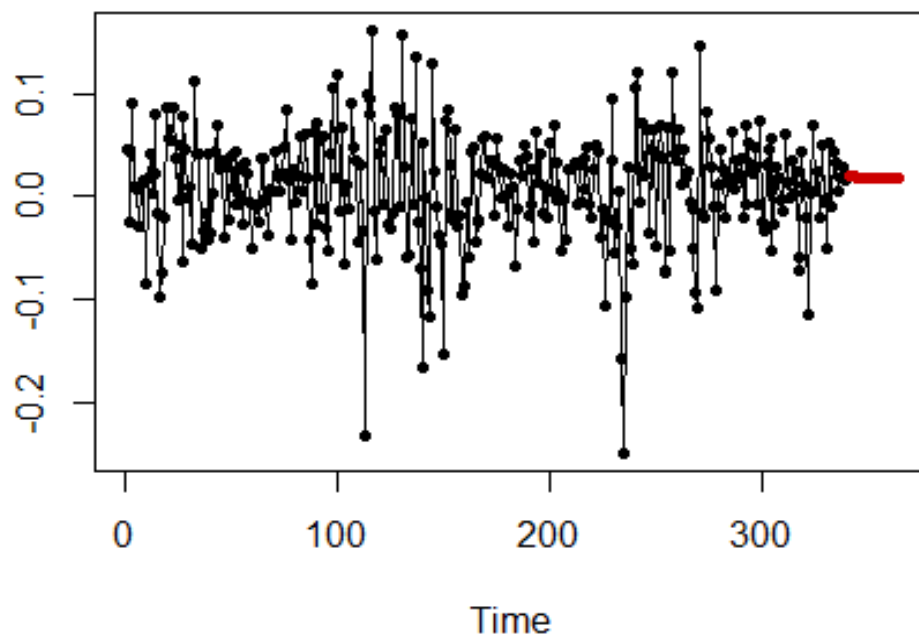


- To Mean Squared Forecast Error, το υπολογίζουμε ως εξής `realdatafcOSGIX <- read.csv2("C:/Users/mihal/OneDrive/realdatafcOSGIX.txt", sep=" ", col.names = "OSGIX")`
`realdatafcOSGIX <- realdatafcOSGIX[,1]`
#OSGIX DATA FROM 8/2017 - 7/2019
`i <- 1:24`
`Y <- realdatafcOSGIX`
`Y_hat <- pred@forecast$seriesFor`
`MSFE <- (sum(Y[i]-Y_hat[i])^2)/24`
`MSFE`

```
## [1] 0.001559561
```

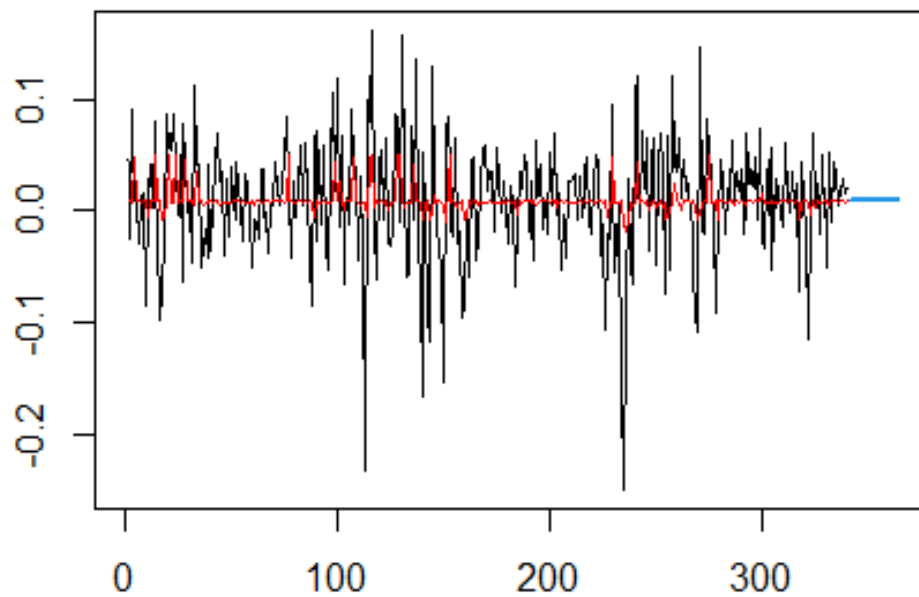
- To Hit Ratio είναι :
- `Hit_ratio <- 19/24`
`Hit_ratio`
- `## [1] 0.7916667`
- *#79% INDICATES A GOOD HIT RATIO*

- Ένας ακόμα τρόπος πρόβλεψης μέσω γενικευμένης παλινδρόμησης νευρωνικών δικτύων (generalized regression neural networks) μας δίνει το εξής :



Ένας τελευταίος τρόπος να πάρουμε μια ιδέα οπτικά για τις προβλεπόμενες αποδόσεις του fund OSGIX μέσω αυτοπαλίνδρομων νευρωνικών δικτύων (Feed-forward neural networks with a single hidden layer and lagged inputs for forecasting univariate time series) είναι :

Forecasts from NNAR(1,4)



Παρατηρούμε πως με εξαίρεση τα νευρωνικά δίκτυα που χρησιμοποιούσαν multiple layer perceptron οι υπόλοιποι τρόποι πρόβλεψης συμφωνούν εν γένει, καταλήγοντας σε θετική πρόβλεψη για την απόδοση του fund OSGIX.