TASK 2

Giuseppina Orefice, Alessandra Campanella

2025-01-12

PART II

a) AgACI paper

The AgACI (Adaptive Aggregated Conformal Inference) extends the ACI method (Adaptive Conformal Inference) by selecting an appropriate adaptation rate γ .

The AgACI method combines the predictions from multiple ACI experts, each with different γ_k using an aggregation strategy, this creates a parameter-free,adaptive method suitable for non-exchangeable data.

We use the AgACI method because the ACI requires a manual tuning of γ and this does not gives us optimal results because the time series change making a single γ inadequate.

With the AgACI multiple experts are introduced, each one corresponds to a different γ_k and combines their prediction dynamically. By aggregating experts outputs this method eliminates the need for manual tuning and ensures the robustness to time series changes.

Each expert k runs an ACI procedure with a specific γ_k chosen form a predefined set $[\gamma_1, \gamma_2, ..., \gamma_k]$. Then AgACI assigns dynamic weights to experts based on their past performance, these weights determine the contribution of each expert to the final prediction interval. Aggregated prediction intervals are typically smaller and more adaptive, maintaining the desired coverage rate $(1 - \alpha)$ over time.

To initialize the AgACI we have to define a set of K learning rates (γ_k) and after that initialize the weights $\omega_k(0) = \frac{1}{K}$ for each expert. Two strategies are proposed based on running ACI for K values, but we are going to focus on the Online Expert Aggregation strategy. This strategy introduce an adaptive aggregation of experts with experts k being ACI with parameters γ_k , then at each time step t, two independent aggregation intervals are computed:

$$\hat{C}_{\alpha k,t} = [\hat{b}^{(l)}_{k,t}, \hat{b}^{(u)}_{k,t}]$$

$$\bar{C}_t = [\bar{b}^{(l)}_t, \bar{b}^{(u)}_t]$$

one for each lower and upper bounds.

This strategy requires a miscoverage error rate α , a grid of values γ, an *aggregation rule* ϕ and some threshold values $M(l)$ and $M(u)$.

Then for every time step t we compute the forecasts using the ACI with every value of γ inside the grid. If one of these forecasts is outside the bounds, then it is replaced with the corresponding threshold value.

After this the weight for every expert are computed based on losses incurred up to time t: $\omega_{t,k} = \phi(\rho_{\beta(.)}(y_s, \hat{b}_{s,l}(x_s)), s \in [T_0 + 1, t], l \in [1, K]$

In this case we use the pinball loss ρ_β , that is common in quantile regression that β is to be chosen between $\alpha/2$ and $1 - \alpha/2$ for lower an upper bound $\beta(l)$ and $\beta(u)$ respectively.

The aggregate forecast is calculated as an optimal weighted mean of the experts' forecasts

$$\bar{b}_{t+1}(x) = \frac{\sum_{k=1}^{K} \omega_{t,k} \hat{b}_{t+1,k}(x)}{\sum_{l=1}^{K} \omega_{t,l}}$$

where the weights assigned to the experts are computed as above. To conclude, the final forecast for each time step is given by aggregating the experts' forecasts.

b) Using the same financial asset chosen in Part I, apply AgACI to quantify the prediction uncertainty of its volatility and evaluate its coverage over time.

point 2.2 and 2.3

Now, it is time to apply the agACI to our model of before. We have chosen to apply it to the apARCH case because: - it was the one that has performed well among the other models; - the paper's author have chosen a quantile regression, but from the theory we know that when we work with the returns we need to model the conditional heteroscedasticity. In effect the GARCH model and its variants are able to model the conditional heteroscedasticity and they can be well applied to the case of the returns of the stocks. We have already explained that there is the clustering of the volatility for the returns and the quantile regression cannot capture at all this kind of behavior. It is useful for estimating conditional quantiles (like value-at-risk), it does not model the changing volatility of the residuals over time. This is a crucial aspect when dealing with financial time series, where understanding and forecasting volatility is key for risk management and asset pricing. In addition, thanks to the GARCH variants like apARCH we are able to achieve more the fact that negative shocks often cause greater volatility than positive shocks of the same magnitude. This flexibility in modeling asymmetric effects is something that quantile regression does not offer in the same way. Even though the GARCH can have some drawbacks, it is still able to model that specific pattern of the returns and that's why our choice is the **GARCH**. We have modified a bit the `garchConformalForecasting` above in order to apply the aggregation function and then the agACI. There are comments inside the box.

```r
library(rugarch)
```

```
## Warning: il pacchetto 'rugarch' è stato creato con R versione 4.3.3
```

```
## Caricamento del pacchetto richiesto: parallel
```

```
## 
## Caricamento pacchetto: 'rugarch'
```

```
## Il seguente oggetto è mascherato da 'package:stats':
## 
##     sigma
```

```r
library(quantmod)
```

```
## Warning: il pacchetto 'quantmod' è stato creato con R versione 4.3.3
```

```
## Caricamento del pacchetto richiesto: xts
```

```
## Warning: il pacchetto 'xts' è stato creato con R versione 4.3.3
```

```
## Caricamento del pacchetto richiesto: zoo
```

```
## Warning: il pacchetto 'zoo' è stato creato con R versione 4.3.3
```

```
## 
## Caricamento pacchetto: 'zoo'
```

```
## I seguenti oggetti sono mascherati da 'package:base':
## 
##     as.Date, as.Date.numeric
```

```
## Caricamento del pacchetto richiesto: TTR
```

```
## Warning: il pacchetto 'TTR' è stato creato con R versione 4.3.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(quantreg)
```

```
## Warning: il pacchetto 'quantreg' è stato creato con R versione 4.3.3
```

```
## Caricamento del pacchetto richiesto: SparseM
```

```
## Warning: il pacchetto 'SparseM' è stato creato con R versione 4.3.3
```

```
## 
## Caricamento pacchetto: 'SparseM'
```

```
## Il seguente oggetto è mascherato da 'package:base':
## 
##     backsolve
```

```
## Warning in .recacheSubclasses(def@className, def, env): sottoclasse non
## definita "ndiMatrix" della classe "replValueSp"; definizione non aggiornata
```

```r
library(zoo)
library(dplyr)
```

```
## Warning: il pacchetto 'dplyr' è stato creato con R versione 4.3.3
```

```
##
## ######################### Warning from 'xts' package ##########################
## #                                                                             #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                            #
## #                                                                             #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                #
## #                                                                             #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                             #
## ###############################################################################
```

```
##
## Caricamento pacchetto: 'dplyr'
```

```
## I seguenti oggetti sono mascherati da 'package:xts':
##
##     first, last
```

```
## I seguenti oggetti sono mascherati da 'package:stats':
##
##     filter, lag
```

```
## I seguenti oggetti sono mascherati da 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(opera)
```

```
## Warning: il pacchetto 'opera' è stato creato con R versione 4.3.3
```

```r
getSymbols("AMZN", from = "2016-12-31", to = "2024-12-31")
```

```
## [1] "AMZN"
```

```r
prices=Cl(AMZN)
returns=diff(log(prices))
returns=na.omit(returns)

tab_gamma_1 = c(0.0004, 0.023)
train_size <- floor(length(returns) * 0.5)
test_size <- length(returns) - train_size
train_data <- returns[1:train_size]
test_data <- returns[(train_size + 1):length(returns)]

garchConformalForecasting_agg <- function(
    returns,
    alpha = 0.10,
    gamma = 0.001,
    lookback = 1250,
    garchP = 1,
    garchQ = 1,
    startUp = 100,
    verbose = FALSE,
    updateMethod = "Momentum",
    momentumBW = 0.95,
    scoreType = "normalized", # Options: "normalized", "non-normalized"
    modelType = "sGARCH",     # Options: "sGARCH", "apARCH"
    forecastSteps = 1,        # Number of forecasting steps ahead
    agg_model = "BOA",        # Aggregation model for weights
    tab_gamma = c(0.0004, 0.023) # Gamma values for the mixture function
) {
  myT <- length(returns)
  T0 <- max(startUp, lookback)
```

```r
# Specify GARCH model based on user input
garchSpec <- ugarchspec(
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  variance.model = list(model = modelType, garchOrder = c(garchP, garchQ)),
  distribution.model = "norm"
)

alphat <- alpha

# Initialize data storage variables
errSeqOC <- rep(0, myT - T0 + 1)
errSeqNC <- rep(0, myT - T0 + 1)
alphaSequence <- rep(alpha, myT - T0 + 1)
scores <- rep(0, myT - T0 + 1)
low_PI <- rep(0, myT - T0 + 1)
high_PI <- rep(0, myT - T0 + 1)
experts_low <- array(NA, dim = c(length(tab_gamma), forecastSteps, test_size))
experts_high <- array(NA, dim = c(length(tab_gamma), forecastSteps, test_size))

for (t in T0:myT) {
  if (verbose) {
    print(t)
  }

  # Fit GARCH model
  garchFit <- ugarchfit(garchSpec, returns[(t - lookback + 1):(t - 1)], solver = "hybrid")
  garchForecast <- ugarchforecast(garchFit, n.ahead = forecastSteps)
  sigmaNext <- sigma(garchForecast)

  # Compute scores based on score type
  for(i in 1:forecastSteps){
    if (scoreType == "normalized") {
      scores[t - T0 + 1] <- abs(returns[t]^2 - sigmaNext[i]^2) / sigmaNext[i]^2
    } else if (scoreType == "non-normalized") {
      scores[t - T0 + 1] <- abs(returns[t]^2 - sigmaNext[i]^2)
    }
  }

  recentScores <- scores[max(t - T0 + 1 - lookback + 1, 1):(t - T0)]

  # Compute errors for both methods
  errSeqOC[t - T0 + 1] <- as.numeric(scores[t - T0 + 1] > quantile(recentScores, 1 - alphat))
  sorted= sort(scores)
  quantile_c= sorted[ceiling((1-alphat)* length(sorted+1))]

  # Update alphat
  alphaSequence[t - T0 + 1] <- alphat
  if (updateMethod == "Simple") {
    alphat <- alphat + gamma * (alpha - errSeqOC[t - T0 + 1])
    alphat <- max(0, min(1, alphat))
  } else if (updateMethod == "Momentum") {
    w <- rev(momentumBW^(1:(t - T0 + 1)))
    w <- w / sum(w)
    alphat <- alphat + gamma * (alpha - sum(errSeqOC[1:(t - T0 + 1)] * w))
    alphat <- max(0, min(1, alphat))
  }

  # Calculate prediction intervals (PI)
  if (scoreType == "non-normalized") {
    low_PI[t - T0 + 1] <- sigmaNext^2 - quantile_c
    high_PI[t - T0 + 1] <- sigmaNext^2 + quantile_c
  } else if (scoreType == "normalized") {
    low_PI[t - T0 + 1] <- sigmaNext^2 - quantile_c * sigmaNext^2
    high_PI[t - T0 + 1] <- sigmaNext^2 + quantile_c * sigmaNext^2
  }

  if (t %% 100 == 0) {
    print(sprintf("Done %g steps", t))
  }
}
# Let's to create the experts
for (idm in seq_along(tab_gamma)) {
  gamma2 <- tab_gamma[idm]
  scale <- sqrt(gamma2)     # We are going to scale the value of gamma
 #in this way we can obtain different predictions for the lower and upper bounds (experts_low and experts_high), effectively controlling the range of the prediction intervals. We can create a set of predictions that span different possible levels of volatility.
  # Compute lower and upper bounds
  experts_low[idm, , ] <- -scale * sigmaNext
```

```
    experts_high[idm, , ] <- scale * sigmaNext
  }


  # The lower bounds are aggregated by using the method mixture; we have chosen the pinball loss to penalize the errors and BOA as aggregation functio
n
  #, which combines these different experts' outputs.
  #After computing predictions from different experts, the mixture function is used to aggregate predictions for the lower (mlpol_grad_low) and upper (mlpol
_grad_high) bounds of the volatility forecast.
  mlpol_grad_low <- mixture(
    Y = as.numeric(test_data),
    experts = aperm(experts_low),
    model = "BOA",
    loss.gradient = TRUE,
    loss.type = list(name = "pinball", tau = 0.05)
  )


  mlpol_grad_high <- mixture(
    Y = as.numeric(test_data),
    experts = aperm(experts_high),
    model = "BOA",
    loss.gradient = TRUE,
    loss.type = list(name = "pinball", tau = 0.95)
  )

  # Aggregated Predictions
  aggregated_low <- mlpol_grad_low$prediction
  aggregated_high <- mlpol_grad_high$prediction


  return(list('Y_inf' = aggregated_low, 'Y_sup' = aggregated_high, 'weights_low' = mlpol_grad_low$weights, 'weights_high' = mlpol_grad_high$weights,
              alphaSequence = alphaSequence, errSeqOC = errSeqOC, sigmaNext=sigmaNext, low_PI = low_PI, high_PI = high_PI, scores=scores, garchFor
ecast=garchForecast))
}
```

During the optimization process, the mixture() function computes the weights that correspond to the importance or contribution of each expert to the final aggregated prediction. These weights are returned as part of the output of the mixture() function. Each prediction was created by using the pinball as measure of accuracy. We have chosen the pinball measure because our primal aim is the identification of a prediction interval and it is a very good measure of the accuracy of the quantile prediction. But we have seen that another good measure can be the log-likelihood loss, that basically is a measure of the accuracy and the efficiency of the entire forecasting. The first application of the function is the GARCH norm.

```
garch_norm_tab1= garchConformalForecasting_agg(
      returns = returns,
      alpha = 0.10,
      gamma = tab_gamma_1,
      modelType = "sGARCH",
      scoreType = "normalized",
      garchP = 1, garchQ = 1,
      startUp = 100,
      verbose = TRUE,
      updateMethod = "Simple",
      momentumBW = 0.95,
      forecastSteps = 5,
      agg_model = "BOA",
      tab_gamma=tab_gamma_1
    )
```

```
## [1] 1250
## [1] 1251
## [1] 1252
## [1] 1253
## [1] 1254
## [1] 1255
## [1] 1256
## [1] 1257
## [1] 1258
## [1] 1259
## [1] 1260
## [1] 1261
## [1] 1262
## [1] 1263
## [1] 1264
## [1] 1265
## [1] 1266
## [1] 1267
## [1] 1268
## [1] 1269
## [1] 1270
## [1] 1271
## [1] 1272
## [1] 1273
## [1] 1274
## [1] 1275
## [1] 1276
## [1] 1277
## [1] 1278
## [1] 1279
## [1] 1280
## [1] 1281
## [1] 1282
## [1] 1283
## [1] 1284
## [1] 1285
## [1] 1286
## [1] 1287
## [1] 1288
## [1] 1289
## [1] 1290
## [1] 1291
## [1] 1292
## [1] 1293
## [1] 1294
## [1] 1295
## [1] 1296
## [1] 1297
## [1] 1298
## [1] 1299
## [1] 1300
## [1] "Done 1300 steps"
## [1] 1301
## [1] 1302
## [1] 1303
## [1] 1304
## [1] 1305
## [1] 1306
## [1] 1307
## [1] 1308
## [1] 1309
## [1] 1310
## [1] 1311
## [1] 1312
## [1] 1313
## [1] 1314
## [1] 1315
## [1] 1316
## [1] 1317
## [1] 1318
## [1] 1319
## [1] 1320
## [1] 1321
## [1] 1322
## [1] 1323
## [1] 1324
## [1] 1325
## [1] 1326
## [1] 1327
## [1] 1328
## [1] 1329
## [1] 1330
## [1] 1331
## [1] 1332
## [1] 1333
## [1] 1334
## [1] 1335
## [1] 1336
## [1] 1337
## [1] 1338
## [1] 1339
## [1] 1340
## [1] 1341
## [1] 1342
## [1] 1343
## [1] 1344
## [1] 1345
## [1] 1346
## [1] 1347
## [1] 1348
## [1] 1349
## [1] 1350
## [1] 1351
## [1] 1352
## [1] 1353
## [1] 1354
## [1] 1355
## [1] 1356
## [1] 1357
## [1] 1358
## [1] 1359
## [1] 1360
## [1] 1361
## [1] 1362
## [1] 1363
## [1] 1364
## [1] 1365
## [1] 1366
## [1] 1367
## [1] 1368
## [1] 1369
## [1] 1370
## [1] 1371
## [1] 1372
## [1] 1373
## [1] 1374
## [1] 1375
## [1] 1376
## [1] 1377
## [1] 1378
## [1] 1379
## [1] 1380
## [1] 1381
## [1] 1382
## [1] 1383
## [1] 1384
## [1] 1385
## [1] 1386
## [1] 1387
## [1] 1388
## [1] 1389
## [1] 1390
## [1] 1391
## [1] 1392
## [1] 1393
## [1] 1394
## [1] 1395
## [1] 1396
## [1] 1397
## [1] 1398
## [1] 1399
## [1] 1400
## [1] "Done 1400 steps"
## [1] 1401
## [1] 1402
## [1] 1403
## [1] 1404
## [1] 1405
## [1] 1406
## [1] 1407
## [1] 1408
## [1] 1409
## [1] 1410
## [1] 1411
## [1] 1412
## [1] 1413
## [1] 1414
## [1] 1415
## [1] 1416
## [1] 1417
## [1] 1418
## [1] 1419
## [1] 1420
## [1] 1421
## [1] 1422
## [1] 1423
## [1] 1424
## [1] 1425
## [1] 1426
## [1] 1427
## [1] 1428
## [1] 1429
## [1] 1430
## [1] 1431
## [1] 1432
## [1] 1433
## [1] 1434
## [1] 1435
## [1] 1436
## [1] 1437
## [1] 1438
## [1] 1439
## [1] 1440
## [1] 1441
## [1] 1442
## [1] 1443
## [1] 1444
## [1] 1445
## [1] 1446
## [1] 1447
## [1] 1448
## [1] 1449
## [1] 1450
## [1] 1451
## [1] 1452
## [1] 1453
## [1] 1454
## [1] 1455
## [1] 1456
## [1] 1457
## [1] 1458
## [1] 1459
## [1] 1460
## [1] 1461
## [1] 1462
## [1] 1463
## [1] 1464
## [1] 1465
## [1] 1466
## [1] 1467
## [1] 1468
## [1] 1469
## [1] 1470
## [1] 1471
## [1] 1472
## [1] 1473
## [1] 1474
## [1] 1475
## [1] 1476
## [1] 1477
## [1] 1478
## [1] 1479
## [1] 1480
## [1] 1481
## [1] 1482
## [1] 1483
## [1] 1484
## [1] 1485
## [1] 1486
## [1] 1487
## [1] 1488
## [1] 1489
## [1] 1490
## [1] 1491
## [1] 1492
## [1] 1493
## [1] 1494
## [1] 1495
## [1] 1496
## [1] 1497
## [1] 1498
## [1] 1499
## [1] 1500
## [1] "Done 1500 steps"
## [1] 1501
## [1] 1502
## [1] 1503
## [1] 1504
## [1] 1505
## [1] 1506
## [1] 1507
## [1] 1508
## [1] 1509
## [1] 1510
## [1] 1511
## [1] 1512
## [1] 1513
## [1] 1514
## [1] 1515
## [1] 1516
## [1] 1517
## [1] 1518
## [1] 1519
## [1] 1520
## [1] 1521
## [1] 1522
## [1] 1523
## [1] 1524
## [1] 1525
## [1] 1526
## [1] 1527
## [1] 1528
## [1] 1529
## [1] 1530
## [1] 1531
## [1] 1532
## [1] 1533
## [1] 1534
## [1] 1535
## [1] 1536
## [1] 1537
## [1] 1538
## [1] 1539
## [1] 1540
## [1] 1541
## [1] 1542
## [1] 1543
## [1] 1544
## [1] 1545
## [1] 1546
## [1] 1547
## [1] 1548
## [1] 1549
## [1] 1550
## [1] 1551
## [1] 1552
## [1] 1553
## [1] 1554
## [1] 1555
## [1] 1556
## [1] 1557
## [1] 1558
## [1] 1559
## [1] 1560
## [1] 1561
## [1] 1562
## [1] 1563
## [1] 1564
## [1] 1565
## [1] 1566
## [1] 1567
## [1] 1568
## [1] 1569
## [1] 1570
## [1] 1571
## [1] 1572
## [1] 1573
## [1] 1574
## [1] 1575
## [1] 1576
## [1] 1577
## [1] 1578
## [1] 1579
## [1] 1580
## [1] 1581
## [1] 1582
## [1] 1583
## [1] 1584
## [1] 1585
## [1] 1586
## [1] 1587
## [1] 1588
## [1] 1589
## [1] 1590
## [1] 1591
## [1] 1592
## [1] 1593
## [1] 1594
## [1] 1595
## [1] 1596
## [1] 1597
## [1] 1598
## [1] 1599
## [1] 1600
## [1] "Done 1600 steps"
## [1] 1601
## [1] 1602
## [1] 1603
## [1] 1604
## [1] 1605
## [1] 1606
## [1] 1607
## [1] 1608
## [1] 1609
## [1] 1610
## [1] 1611
## [1] 1612
## [1] 1613
## [1] 1614
## [1] 1615
## [1] 1616
## [1] 1617
## [1] 1618
## [1] 1619
## [1] 1620
## [1] 1621
## [1] 1622
## [1] 1623
## [1] 1624
## [1] 1625
## [1] 1626
## [1] 1627
## [1] 1628
## [1] 1629
## [1] 1630
## [1] 1631
## [1] 1632
## [1] 1633
## [1] 1634
## [1] 1635
## [1] 1636
## [1] 1637
## [1] 1638
## [1] 1639
## [1] 1640
## [1] 1641
## [1] 1642
## [1] 1643
## [1] 1644
## [1] 1645
## [1] 1646
## [1] 1647
## [1] 1648
## [1] 1649
## [1] 1650
## [1] 1651
## [1] 1652
## [1] 1653
## [1] 1654
## [1] 1655
## [1] 1656
## [1] 1657
## [1] 1658
## [1] 1659
## [1] 1660
## [1] 1661
## [1] 1662
## [1] 1663
## [1] 1664
## [1] 1665
## [1] 1666
## [1] 1667
## [1] 1668
## [1] 1669
## [1] 1670
## [1] 1671
## [1] 1672
## [1] 1673
## [1] 1674
## [1] 1675
## [1] 1676
## [1] 1677
## [1] 1678
## [1] 1679
## [1] 1680
## [1] 1681
## [1] 1682
## [1] 1683
## [1] 1684
## [1] 1685
## [1] 1686
## [1] 1687
## [1] 1688
## [1] 1689
## [1] 1690
## [1] 1691
## [1] 1692
## [1] 1693
## [1] 1694
## [1] 1695
## [1] 1696
## [1] 1697
## [1] 1698
## [1] 1699
## [1] 1700
## [1] "Done 1700 steps"
## [1] 1701
## [1] 1702
## [1] 1703
## [1] 1704
## [1] 1705
## [1] 1706
## [1] 1707
## [1] 1708
## [1] 1709
## [1] 1710
## [1] 1711
## [1] 1712
## [1] 1713
## [1] 1714
## [1] 1715
## [1] 1716
## [1] 1717
## [1] 1718
## [1] 1719
## [1] 1720
## [1] 1721
## [1] 1722
## [1] 1723
## [1] 1724
## [1] 1725
## [1] 1726
## [1] 1727
## [1] 1728
## [1] 1729
## [1] 1730
## [1] 1731
## [1] 1732
## [1] 1733
## [1] 1734
## [1] 1735
## [1] 1736
## [1] 1737
## [1] 1738
## [1] 1739
## [1] 1740
## [1] 1741
## [1] 1742
## [1] 1743
## [1] 1744
## [1] 1745
## [1] 1746
## [1] 1747
## [1] 1748
## [1] 1749
## [1] 1750
## [1] 1751
## [1] 1752
## [1] 1753
## [1] 1754
## [1] 1755
## [1] 1756
## [1] 1757
## [1] 1758
## [1] 1759
## [1] 1760
## [1] 1761
## [1] 1762
## [1] 1763
## [1] 1764
## [1] 1765
## [1] 1766
## [1] 1767
## [1] 1768
## [1] 1769
## [1] 1770
## [1] 1771
## [1] 1772
## [1] 1773
## [1] 1774
## [1] 1775
## [1] 1776
## [1] 1777
## [1] 1778
## [1] 1779
## [1] 1780
## [1] 1781
## [1] 1782
## [1] 1783
## [1] 1784
## [1] 1785
## [1] 1786
## [1] 1787
## [1] 1788
## [1] 1789
## [1] 1790
## [1] 1791
## [1] 1792
## [1] 1793
## [1] 1794
## [1] 1795
## [1] 1796
## [1] 1797
## [1] 1798
## [1] 1799
## [1] 1800
## [1] "Done 1800 steps"
## [1] 1801
## [1] 1802
## [1] 1803
## [1] 1804
## [1] 1805
## [1] 1806
## [1] 1807
## [1] 1808
## [1] 1809
## [1] 1810
## [1] 1811
## [1] 1812
## [1] 1813
## [1] 1814
## [1] 1815
## [1] 1816
## [1] 1817
## [1] 1818
## [1] 1819
## [1] 1820
## [1] 1821
## [1] 1822
## [1] 1823
## [1] 1824
## [1] 1825
## [1] 1826
## [1] 1827
## [1] 1828
## [1] 1829
## [1] 1830
## [1] 1831
## [1] 1832
## [1] 1833
## [1] 1834
## [1] 1835
## [1] 1836
## [1] 1837
## [1] 1838
## [1] 1839
## [1] 1840
## [1] 1841
## [1] 1842
## [1] 1843
## [1] 1844
## [1] 1845
## [1] 1846
## [1] 1847
## [1] 1848
## [1] 1849
## [1] 1850
## [1] 1851
## [1] 1852
## [1] 1853
## [1] 1854
## [1] 1855
## [1] 1856
## [1] 1857
## [1] 1858
## [1] 1859
## [1] 1860
## [1] 1861
## [1] 1862
## [1] 1863
## [1] 1864
## [1] 1865
## [1] 1866
## [1] 1867
## [1] 1868
## [1] 1869
## [1] 1870
## [1] 1871
## [1] 1872
## [1] 1873
## [1] 1874
## [1] 1875
## [1] 1876
## [1] 1877
## [1] 1878
## [1] 1879
## [1] 1880
## [1] 1881
## [1] 1882
## [1] 1883
## [1] 1884
## [1] 1885
## [1] 1886
## [1] 1887
## [1] 1888
## [1] 1889
## [1] 1890
## [1] 1891
## [1] 1892
## [1] 1893
## [1] 1894
## [1] 1895
## [1] 1896
## [1] 1897
## [1] 1898
## [1] 1899
## [1] 1900
## [1] "Done 1900 steps"
## [1] 1901
## [1] 1902
## [1] 1903
## [1] 1904
## [1] 1905
## [1] 1906
## [1] 1907
## [1] 1908
## [1] 1909
## [1] 1910
## [1] 1911
## [1] 1912
## [1] 1913
## [1] 1914
## [1] 1915
## [1] 1916
## [1] 1917
## [1] 1918
## [1] 1919
## [1] 1920
## [1] 1921
## [1] 1922
## [1] 1923
## [1] 1924
## [1] 1925
## [1] 1926
## [1] 1927
## [1] 1928
## [1] 1929
## [1] 1930
## [1] 1931
## [1] 1932
## [1] 1933
## [1] 1934
## [1] 1935
## [1] 1936
## [1] 1937
## [1] 1938
## [1] 1939
## [1] 1940
## [1] 1941
## [1] 1942
## [1] 1943
## [1] 1944
## [1] 1945
## [1] 1946
## [1] 1947
## [1] 1948
## [1] 1949
## [1] 1950
## [1] 1951
## [1] 1952
## [1] 1953
## [1] 1954
## [1] 1955
## [1] 1956
## [1] 1957
## [1] 1958
## [1] 1959
## [1] 1960
## [1] 1961
## [1] 1962
## [1] 1963
## [1] 1964
## [1] 1965
## [1] 1966
## [1] 1967
## [1] 1968
## [1] 1969
## [1] 1970
## [1] 1971
## [1] 1972
## [1] 1973
## [1] 1974
## [1] 1975
## [1] 1976
## [1] 1977
## [1] 1978
## [1] 1979
## [1] 1980
## [1] 1981
## [1] 1982
## [1] 1983
## [1] 1984
## [1] 1985
## [1] 1986
## [1] 1987
## [1] 1988
## [1] 1989
## [1] 1990
## [1] 1991
## [1] 1992
## [1] 1993
## [1] 1994
## [1] 1995
## [1] 1996
## [1] 1997
## [1] 1998
## [1] 1999
## [1] 2000
## [1] "Done 2000 steps"
## [1] 2001
## [1] 2002
## [1] 2003
## [1] 2004
## [1] 2005
## [1] 2006
## [1] 2007
## [1] 2008
## [1] 2009
## [1] 2010
```

```
tab_gamma_2=c(0.01, 0.088, 0.0093, 0.01)
garch_norm_tab2= garchConformalForecasting_agg(
    returns = returns,
    alpha = 0.10,
    gamma = tab_gamma_2,
    modelType = "sGARCH",
    scoreType = "normalized",
    garchP = 1, garchQ = 1,
    startUp = 100,
    verbose = TRUE,
    updateMethod = "Simple",
    momentumBW = 0.95,
    forecastSteps = 5,
    agg_model = "BOA",
    tab_gamma=tab_gamma_2
  )
```

```
## [1] 1250
## [1] 1251
## [1] 1252
## [1] 1253
## [1] 1254
## [1] 1255
## [1] 1256
## [1] 1257
## [1] 1258
## [1] 1259
## [1] 1260
## [1] 1261
## [1] 1262
## [1] 1263
## [1] 1264
## [1] 1265
## [1] 1266
## [1] 1267
## [1] 1268
## [1] 1269
## [1] 1270
## [1] 1271
## [1] 1272
## [1] 1273
## [1] 1274
## [1] 1275
## [1] 1276
## [1] 1277
## [1] 1278
## [1] 1279
## [1] 1280
## [1] 1281
## [1] 1282
## [1] 1283
## [1] 1284
## [1] 1285
## [1] 1286
## [1] 1287
## [1] 1288
## [1] 1289
## [1] 1290
## [1] 1291
## [1] 1292
## [1] 1293
## [1] 1294
## [1] 1295
## [1] 1296
## [1] 1297
## [1] 1298
## [1] 1299
## [1] 1300
## [1] "Done 1300 steps"
## [1] 1301
## [1] 1302
## [1] 1303
## [1] 1304
## [1] 1305
## [1] 1306
## [1] 1307
## [1] 1308
## [1] 1309
## [1] 1310
## [1] 1311
## [1] 1312
## [1] 1313
## [1] 1314
## [1] 1315
## [1] 1316
## [1] 1317
## [1] 1318
## [1] 1319
## [1] 1320
## [1] 1321
## [1] 1322
## [1] 1323
## [1] 1324
## [1] 1325
## [1] 1326
## [1] 1327
## [1] 1328
## [1] 1329
## [1] 1330
## [1] 1331
## [1] 1332
## [1] 1333
## [1] 1334
## [1] 1335
## [1] 1336
## [1] 1337
## [1] 1338
## [1] 1339
## [1] 1340
## [1] 1341
## [1] 1342
## [1] 1343
## [1] 1344
## [1] 1345
## [1] 1346
## [1] 1347
## [1] 1348
## [1] 1349
## [1] 1350
## [1] 1351
## [1] 1352
## [1] 1353
## [1] 1354
## [1] 1355
## [1] 1356
## [1] 1357
## [1] 1358
## [1] 1359
## [1] 1360
## [1] 1361
## [1] 1362
## [1] 1363
## [1] 1364
## [1] 1365
## [1] 1366
## [1] 1367
## [1] 1368
## [1] 1369
## [1] 1370
## [1] 1371
## [1] 1372
## [1] 1373
## [1] 1374
## [1] 1375
## [1] 1376
## [1] 1377
## [1] 1378
## [1] 1379
## [1] 1380
## [1] 1381
## [1] 1382
## [1] 1383
## [1] 1384
## [1] 1385
## [1] 1386
## [1] 1387
## [1] 1388
## [1] 1389
## [1] 1390
## [1] 1391
## [1] 1392
## [1] 1393
## [1] 1394
## [1] 1395
## [1] 1396
## [1] 1397
## [1] 1398
## [1] 1399
## [1] 1400
## [1] "Done 1400 steps"
## [1] 1401
## [1] 1402
## [1] 1403
## [1] 1404
## [1] 1405
## [1] 1406
## [1] 1407
## [1] 1408
## [1] 1409
## [1] 1410
## [1] 1411
## [1] 1412
## [1] 1413
## [1] 1414
## [1] 1415
## [1] 1416
## [1] 1417
## [1] 1418
## [1] 1419
## [1] 1420
## [1] 1421
## [1] 1422
## [1] 1423
## [1] 1424
## [1] 1425
## [1] 1426
## [1] 1427
## [1] 1428
## [1] 1429
## [1] 1430
## [1] 1431
## [1] 1432
## [1] 1433
## [1] 1434
## [1] 1435
## [1] 1436
## [1] 1437
## [1] 1438
## [1] 1439
## [1] 1440
## [1] 1441
## [1] 1442
## [1] 1443
## [1] 1444
## [1] 1445
## [1] 1446
## [1] 1447
## [1] 1448
## [1] 1449
## [1] 1450
## [1] 1451
## [1] 1452
## [1] 1453
## [1] 1454
## [1] 1455
## [1] 1456
## [1] 1457
## [1] 1458
## [1] 1459
## [1] 1460
## [1] 1461
## [1] 1462
## [1] 1463
## [1] 1464
## [1] 1465
## [1] 1466
## [1] 1467
## [1] 1468
## [1] 1469
## [1] 1470
## [1] 1471
## [1] 1472
## [1] 1473
## [1] 1474
## [1] 1475
## [1] 1476
## [1] 1477
## [1] 1478
## [1] 1479
## [1] 1480
## [1] 1481
## [1] 1482
## [1] 1483
## [1] 1484
## [1] 1485
## [1] 1486
## [1] 1487
## [1] 1488
## [1] 1489
## [1] 1490
## [1] 1491
## [1] 1492
## [1] 1493
## [1] 1494
## [1] 1495
## [1] 1496
## [1] 1497
## [1] 1498
## [1] 1499
## [1] 1500
## [1] "Done 1500 steps"
## [1] 1501
## [1] 1502
## [1] 1503
## [1] 1504
## [1] 1505
## [1] 1506
## [1] 1507
## [1] 1508
## [1] 1509
## [1] 1510
## [1] 1511
## [1] 1512
## [1] 1513
## [1] 1514
## [1] 1515
## [1] 1516
## [1] 1517
## [1] 1518
## [1] 1519
## [1] 1520
## [1] 1521
## [1] 1522
## [1] 1523
## [1] 1524
## [1] 1525
## [1] 1526
## [1] 1527
## [1] 1528
## [1] 1529
## [1] 1530
## [1] 1531
## [1] 1532
## [1] 1533
## [1] 1534
## [1] 1535
## [1] 1536
## [1] 1537
## [1] 1538
## [1] 1539
## [1] 1540
## [1] 1541
## [1] 1542
## [1] 1543
## [1] 1544
## [1] 1545
## [1] 1546
## [1] 1547
## [1] 1548
## [1] 1549
## [1] 1550
## [1] 1551
## [1] 1552
## [1] 1553
## [1] 1554
## [1] 1555
## [1] 1556
## [1] 1557
## [1] 1558
## [1] 1559
## [1] 1560
## [1] 1561
## [1] 1562
## [1] 1563
## [1] 1564
## [1] 1565
## [1] 1566
## [1] 1567
## [1] 1568
## [1] 1569
## [1] 1570
## [1] 1571
## [1] 1572
## [1] 1573
## [1] 1574
## [1] 1575
## [1] 1576
## [1] 1577
## [1] 1578
## [1] 1579
## [1] 1580
## [1] 1581
## [1] 1582
## [1] 1583
## [1] 1584
## [1] 1585
## [1] 1586
## [1] 1587
## [1] 1588
## [1] 1589
## [1] 1590
## [1] 1591
## [1] 1592
## [1] 1593
## [1] 1594
## [1] 1595
## [1] 1596
## [1] 1597
## [1] 1598
## [1] 1599
## [1] 1600
## [1] "Done 1600 steps"
## [1] 1601
## [1] 1602
## [1] 1603
## [1] 1604
## [1] 1605
## [1] 1606
## [1] 1607
## [1] 1608
## [1] 1609
## [1] 1610
## [1] 1611
## [1] 1612
## [1] 1613
## [1] 1614
## [1] 1615
## [1] 1616
## [1] 1617
## [1] 1618
## [1] 1619
## [1] 1620
## [1] 1621
## [1] 1622
## [1] 1623
## [1] 1624
## [1] 1625
## [1] 1626
## [1] 1627
## [1] 1628
## [1] 1629
## [1] 1630
## [1] 1631
## [1] 1632
## [1] 1633
## [1] 1634
## [1] 1635
## [1] 1636
## [1] 1637
## [1] 1638
## [1] 1639
## [1] 1640
## [1] 1641
## [1] 1642
## [1] 1643
## [1] 1644
## [1] 1645
## [1] 1646
## [1] 1647
## [1] 1648
## [1] 1649
## [1] 1650
## [1] 1651
## [1] 1652
## [1] 1653
## [1] 1654
## [1] 1655
## [1] 1656
## [1] 1657
## [1] 1658
## [1] 1659
## [1] 1660
## [1] 1661
## [1] 1662
## [1] 1663
## [1] 1664
## [1] 1665
## [1] 1666
## [1] 1667
## [1] 1668
## [1] 1669
## [1] 1670
## [1] 1671
## [1] 1672
## [1] 1673
## [1] 1674
## [1] 1675
## [1] 1676
## [1] 1677
## [1] 1678
## [1] 1679
## [1] 1680
## [1] 1681
## [1] 1682
## [1] 1683
## [1] 1684
## [1] 1685
## [1] 1686
## [1] 1687
## [1] 1688
## [1] 1689
## [1] 1690
## [1] 1691
## [1] 1692
## [1] 1693
## [1] 1694
## [1] 1695
## [1] 1696
## [1] 1697
## [1] 1698
## [1] 1699
## [1] 1700
## [1] "Done 1700 steps"
## [1] 1701
## [1] 1702
## [1] 1703
## [1] 1704
## [1] 1705
## [1] 1706
## [1] 1707
## [1] 1708
## [1] 1709
## [1] 1710
## [1] 1711
## [1] 1712
## [1] 1713
## [1] 1714
## [1] 1715
## [1] 1716
## [1] 1717
## [1] 1718
## [1] 1719
## [1] 1720
## [1] 1721
## [1] 1722
## [1] 1723
## [1] 1724
## [1] 1725
## [1] 1726
## [1] 1727
## [1] 1728
## [1] 1729
## [1] 1730
## [1] 1731
## [1] 1732
## [1] 1733
## [1] 1734
## [1] 1735
## [1] 1736
## [1] 1737
## [1] 1738
## [1] 1739
## [1] 1740
## [1] 1741
## [1] 1742
## [1] 1743
## [1] 1744
## [1] 1745
## [1] 1746
## [1] 1747
## [1] 1748
## [1] 1749
## [1] 1750
## [1] 1751
## [1] 1752
## [1] 1753
## [1] 1754
## [1] 1755
## [1] 1756
## [1] 1757
## [1] 1758
## [1] 1759
## [1] 1760
## [1] 1761
## [1] 1762
## [1] 1763
## [1] 1764
## [1] 1765
## [1] 1766
## [1] 1767
## [1] 1768
## [1] 1769
## [1] 1770
## [1] 1771
## [1] 1772
## [1] 1773
## [1] 1774
## [1] 1775
## [1] 1776
## [1] 1777
## [1] 1778
## [1] 1779
## [1] 1780
## [1] 1781
## [1] 1782
## [1] 1783
## [1] 1784
## [1] 1785
## [1] 1786
## [1] 1787
## [1] 1788
## [1] 1789
## [1] 1790
## [1] 1791
## [1] 1792
## [1] 1793
## [1] 1794
## [1] 1795
## [1] 1796
## [1] 1797
## [1] 1798
## [1] 1799
## [1] 1800
## [1] "Done 1800 steps"
## [1] 1801
## [1] 1802
## [1] 1803
## [1] 1804
## [1] 1805
## [1] 1806
## [1] 1807
## [1] 1808
## [1] 1809
## [1] 1810
## [1] 1811
## [1] 1812
## [1] 1813
## [1] 1814
## [1] 1815
## [1] 1816
## [1] 1817
## [1] 1818
## [1] 1819
## [1] 1820
## [1] 1821
## [1] 1822
## [1] 1823
## [1] 1824
## [1] 1825
## [1] 1826
## [1] 1827
## [1] 1828
## [1] 1829
## [1] 1830
## [1] 1831
## [1] 1832
## [1] 1833
## [1] 1834
## [1] 1835
## [1] 1836
## [1] 1837
## [1] 1838
## [1] 1839
## [1] 1840
## [1] 1841
## [1] 1842
## [1] 1843
## [1] 1844
## [1] 1845
## [1] 1846
## [1] 1847
## [1] 1848
## [1] 1849
## [1] 1850
## [1] 1851
## [1] 1852
## [1] 1853
## [1] 1854
## [1] 1855
## [1] 1856
## [1] 1857
## [1] 1858
## [1] 1859
## [1] 1860
## [1] 1861
## [1] 1862
## [1] 1863
## [1] 1864
## [1] 1865
## [1] 1866
## [1] 1867
## [1] 1868
## [1] 1869
## [1] 1870
## [1] 1871
## [1] 1872
## [1] 1873
## [1] 1874
## [1] 1875
## [1] 1876
## [1] 1877
## [1] 1878
## [1] 1879
## [1] 1880
## [1] 1881
## [1] 1882
## [1] 1883
## [1] 1884
## [1] 1885
## [1] 1886
## [1] 1887
## [1] 1888
## [1] 1889
## [1] 1890
## [1] 1891
## [1] 1892
## [1] 1893
## [1] 1894
## [1] 1895
## [1] 1896
## [1] 1897
## [1] 1898
## [1] 1899
## [1] 1900
## [1] "Done 1900 steps"
## [1] 1901
## [1] 1902
## [1] 1903
## [1] 1904
## [1] 1905
## [1] 1906
## [1] 1907
## [1] 1908
## [1] 1909
## [1] 1910
## [1] 1911
## [1] 1912
## [1] 1913
## [1] 1914
## [1] 1915
## [1] 1916
## [1] 1917
## [1] 1918
## [1] 1919
## [1] 1920
## [1] 1921
## [1] 1922
## [1] 1923
## [1] 1924
## [1] 1925
## [1] 1926
## [1] 1927
## [1] 1928
## [1] 1929
## [1] 1930
## [1] 1931
## [1] 1932
## [1] 1933
## [1] 1934
## [1] 1935
## [1] 1936
## [1] 1937
## [1] 1938
## [1] 1939
## [1] 1940
## [1] 1941
## [1] 1942
## [1] 1943
## [1] 1944
## [1] 1945
## [1] 1946
## [1] 1947
## [1] 1948
## [1] 1949
## [1] 1950
## [1] 1951
## [1] 1952
## [1] 1953
## [1] 1954
## [1] 1955
## [1] 1956
## [1] 1957
## [1] 1958
## [1] 1959
## [1] 1960
## [1] 1961
## [1] 1962
## [1] 1963
## [1] 1964
## [1] 1965
## [1] 1966
## [1] 1967
## [1] 1968
## [1] 1969
## [1] 1970
## [1] 1971
## [1] 1972
## [1] 1973
## [1] 1974
## [1] 1975
## [1] 1976
## [1] 1977
## [1] 1978
## [1] 1979
## [1] 1980
## [1] 1981
## [1] 1982
## [1] 1983
## [1] 1984
## [1] 1985
## [1] 1986
## [1] 1987
## [1] 1988
## [1] 1989
## [1] 1990
## [1] 1991
## [1] 1992
## [1] 1993
## [1] 1994
## [1] 1995
## [1] 1996
## [1] 1997
## [1] 1998
## [1] 1999
## [1] 2000
## [1] "Done 2000 steps"
## [1] 2001
## [1] 2002
## [1] 2003
## [1] 2004
## [1] 2005
## [1] 2006
## [1] 2007
## [1] 2008
## [1] 2009
## [1] 2010
```

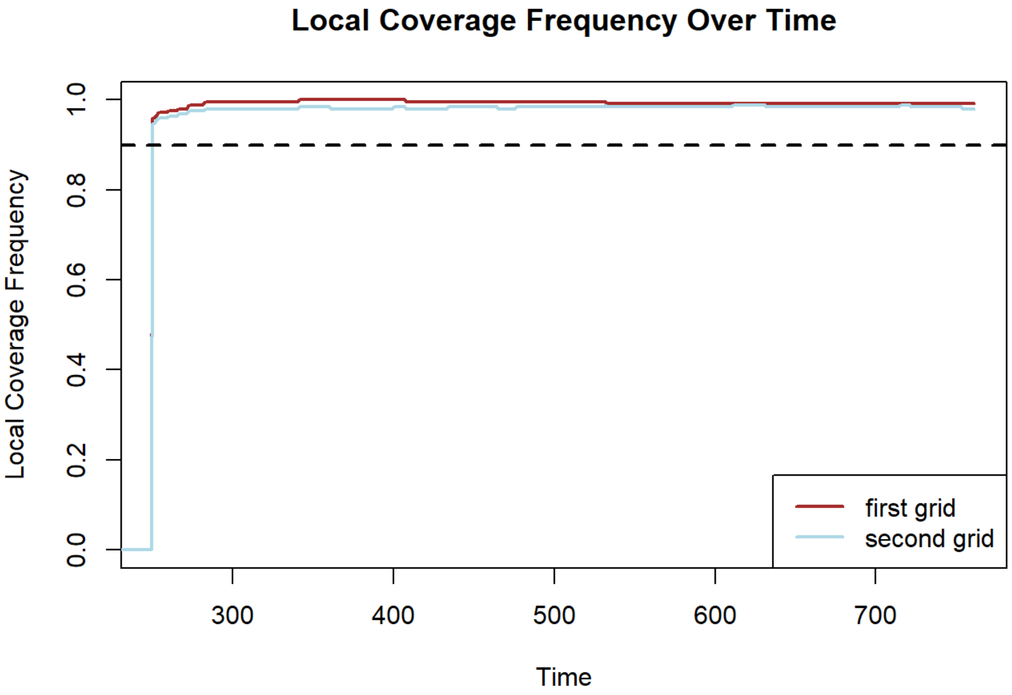
As aggregation model we have chosen **BOA (Bernstein Online Aggregation)**. The BOA algorithm often achieves better generalization performance compared to individual models due to its iterative correction of errors, which helps reduce bias and variance in predictions. It means that it handles overfitting and underfitting adjusting the weights dynamically and automatically. In addition, the BOA is robust to the change in distribution that occurs frequently in this framework, but it is very expensive. Other models are **exponentially weighted average** and the **online gradient descent**.They are both flexible and not expensive, but the first one, the EWA can give more weight to the recent observations with the respect the past observations. This could be also reasonable, because we care a lot of the recent observations and not the far observations! The main drawback is the overfitting and in addition the estimation of the learning rate: if the learning rate is too high, it could lead to instability in predictions, which might be problematic for the highly volatile nature of financial data in GARCH models. The OGD can dynamically adjust the learning rate and is suited for problems with continuous updates. It could adapt well to the time-varying nature of volatility and is robust to changes in the input data, but also in this case the hyperparameter tuning is expensive and we should pay attention to the calibration that we are going to make. Let’s plot the coverage rate through time and comparing it with the apARCH

```r
calculateLocalCoverage <- function(errSeqOC, windowSize = 250) {
  n <- length(errSeqOC)
  localCoverage <- numeric(n)

  for (t in seq(windowSize, n)) {
    localCoverage[t] <- 1 - sum(errSeqOC[(t - windowSize + 1):t]) / windowSize
  }

  return(localCoverage)
}
# Apply the function
localCovt_garch_norm_tab1 <- calculateLocalCoverage(garch_norm_tab1$errSeqOC, windowSize = 250)
localCovt_garch_norm_tab2 <- calculateLocalCoverage(garch_norm_tab2$errSeqOC, windowSize = 250)
target_value= 1-0.10

# Plot the results (optional)
plot(localCovt_garch_norm_tab1, col = "brown", type = "l",
     main = "Local Coverage Frequency Over Time",
     xlab = "Time", ylab = "Local Coverage Frequency",
     lwd = 2, xlim = c(251, length(garch_norm_tab1$errSeqOC)))
lines(localCovt_garch_norm_tab2, col = "lightblue", lwd = 2)
abline(h = target_value, col = "black", lty = 2, lwd = 2)
legend("bottomright",
       legend = c("first grid", "second grid"),
       col = c("brown", "lightblue","magenta"),
       lwd = c(2, 2), lty = c(1, 1))
```



As we can see from the local coverage plot, the agACI, when applied with both the grid of the K experts, demonstrates a significantly overcoverage over time compared to the ACI technique applied to the apARCH model, using the best gamma value calculated earlier. It basically means that agACI tends to overestimate the interval and it is not so much resilient to fluctuations in the market data. By aggregating the predictions from multiple experts with varying volatility assumptions, the agACI method maybe offers a more reliable and robust forecast, but it does not improves so much the overall quality and precision of the prediction intervals in terms of coverage rate. Sharp deviations above or below 0.90 indicate periods where the coverage rate is significantly different from the target and in the plot above this happened all over the time.