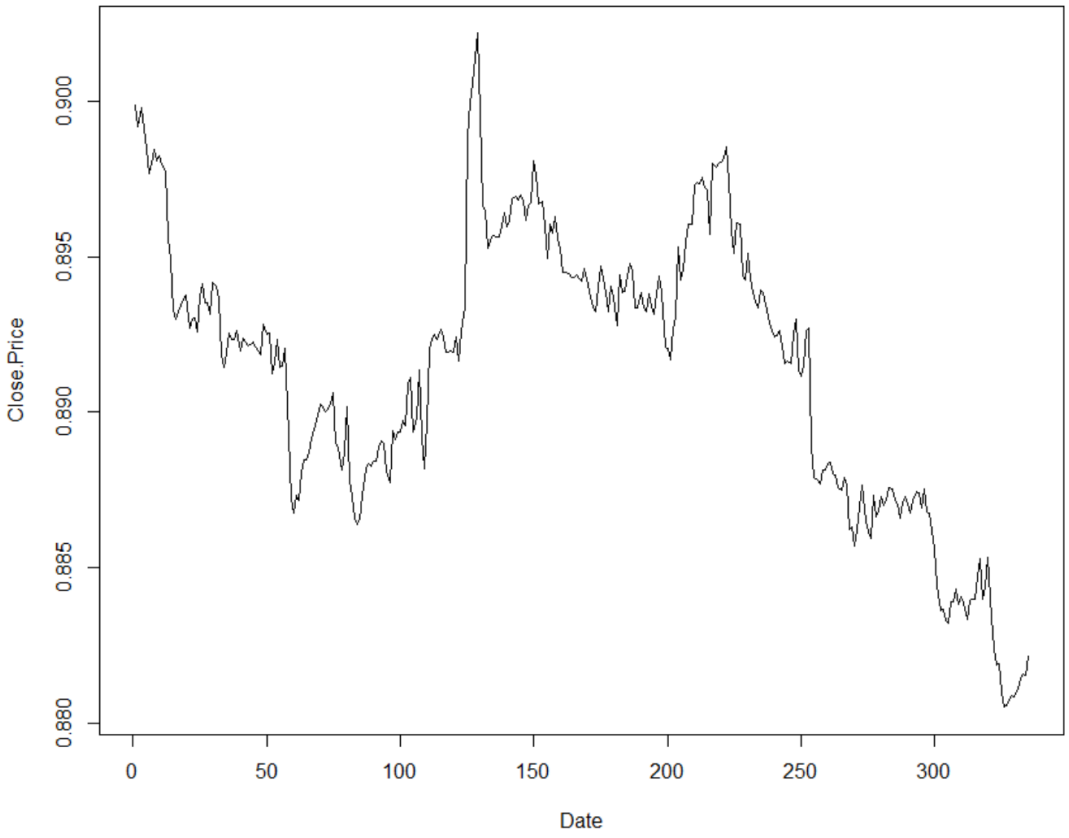
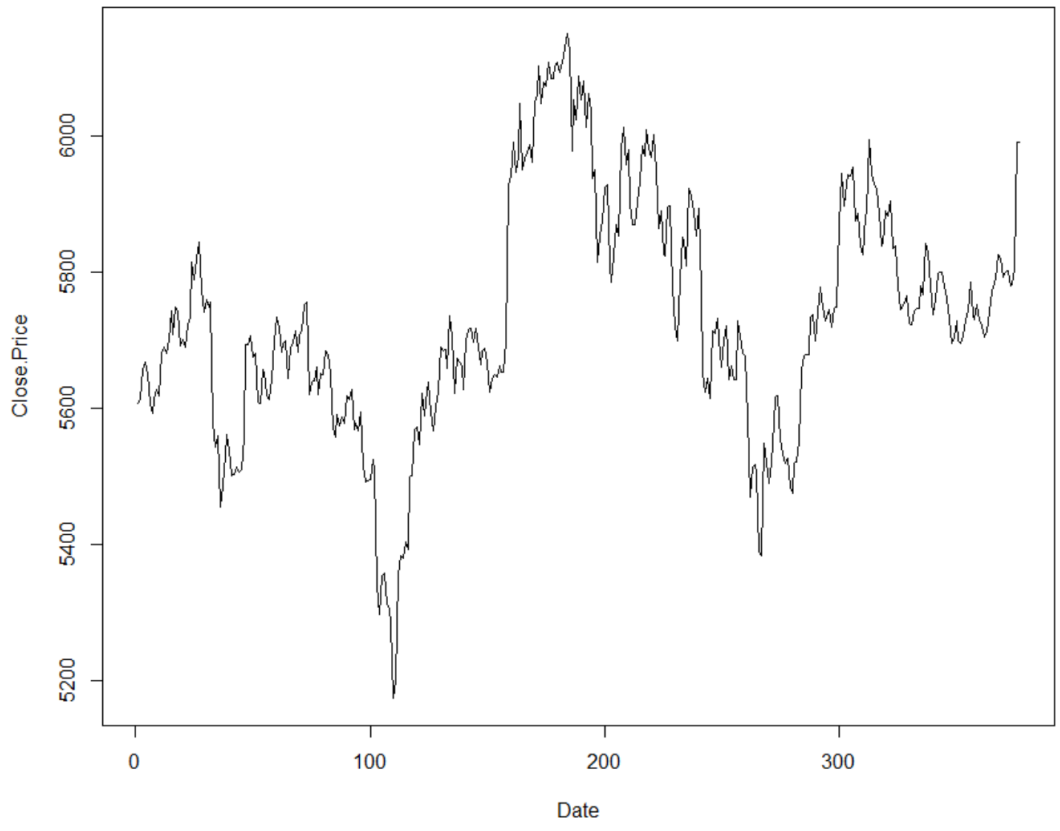
Exercise 7 - Multi-Model Comparison of Cryptocurrency vs Forex (in R)

November 22, 2017

Note: in all predictions below the same FTS data is used (400h EUR/GBP October 2017 & 400h BTC/USD October 2017)





1. AR Yule-Walker model predictions

Forex (EUR/GBP) accuracy:
1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20.
0.52 0.52 0.51 0.51 0.51 0.53 0.51 0.50 0.50 0.49 0.49 0.50 0.48 0.49 0.48 0.49 0.48 0.49 0.50 0.50, $\mu = 0.50$.
(234 predictions)

Cryptocurrency (BTC/USD) accuracy:
1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20.
0.50 0.50 0.53 0.53 0.54 0.54 0.54 0.55 0.54 0.54 0.54 0.55 0.54 0.53 0.53 0.53 0.53 0.54 0.54 0.55, $\mu = 0.53$.
(276 predictions)

Script used:

```
#Author: Thomas Hollis
#Subject: Bachelor Thesis

#1. Data Import
data <- read.csv("R/BTC.csv")

#2. Data Split
data_train <- data[1:100,2]
data_test <- data[101:377,2]

#3. Model Train & apply Model
performance <- 0
for (j in 1:20)
{
  data_train <- data[1:100,2]
  c_count <- 0
  for (i in 1:277)
  {
    btc_ar <- ar.yw(data_train, order.max = j, aic = FALSE)
    btc_pred <- predict(object = btc_ar, n.ahead = 1)

    if(i > 1)
    {
      if(((btc_pred$pred[1] > data_test[i-1]) && (data_test[i] > data_test[i-1])) ||
((btc_pred$pred[1] < data_test[i-1]) && (data_test[i] < data_test[i-1])))
      {
        c_count = c_count+1
      }
    }

    data_train <- c(data_train, data_test[i])
  }
  performance[j] <- c_count/276.00
}
round(performance, digits = 2)
round(mean(performance), digits =2)
```

2. AR Burg model predictions

Forex (EUR/GBP) accuracy:
1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20.
0.52 0.52 0.53 0.52 0.52 0.53 0.51 0.50 0.50 0.50 0.50 0.49 0.48 0.52 0.47 0.47 0.48 0.47 0.47 0.47, $\mu = 0.50$ (234 predictions)

Cryptocurrency (BTC/USD) accuracy:
1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20.
0.50 0.50 0.54 0.54 0.54 0.52 0.53 0.51 0.53 0.51 0.52 0.54 0.54 0.53 0.53 0.51 0.50 0.54 0.54 0.53, $\mu = 0.52$. (276 predictions)

Script used:

```
#Author: Thomas Hollis
#Subject: Bachelor Thesis

#1. Data Import
data <- read.csv("R/BTC.csv")

#2. Data Split
data_train <- data[1:100,2]
data_test <- data[101:377,2]

#3. Model Train & apply Model
performance <- 0
for (j in 1:20)
{
  data_train <- data[1:100,2]
  c_count <- 0
  for (i in 1:277)
  {
    btc_ar <- ar.burg(data_train, order.max = j, aic = FALSE)
    btc_pred <- predict(object = btc_ar, n.ahead = 1)

    if(i > 1)
    {
      if(((btc_pred$pred[1] > data_test[i-1]) && (data_test[i] > data_test[i-1])) ||
((btc_pred$pred[1] < data_test[i-1]) && (data_test[i] < data_test[i-1])))
      {
        c_count = c_count+1
      }
    }

    data_train <- c(data_train, data_test[i])
  }
  performance[j] <- c_count/276.00
}
round(performance, digits = 2)
round(mean(performance), digits =2)
```

3. ARMA model predictions

Forex (EUR/GBP) accuracy:

404.0	204.00	502.00	306.00	103.00	501.00	606.00	103.00	206.0	503.00	405.00	405.00	405.00	504.00	404.0	402.0	403.00	603.00	505.00	302.0
0.5	0.53	0.52	0.49	0.52	0.55	0.52	0.52	0.5	0.54	0.48	0.48	0.48	0.49	0.5	0.5	0.51	0.55	0.49	0.5

$\mu = 0.51$
(234 predictions)

Cryptocurrency (BTC/USD) accuracy:

401.00	503.00	505.00	404.00	205.00	505.00	202.0	405.00	103.00	501.00	103.00	305.00	303.00	304.00	503.00	302.00	406.0	303.00	302.00	502.00
0.53	0.48	0.54	0.51	0.56	0.54	0.5	0.51	0.53	0.51	0.53	0.54	0.51	0.53	0.48	0.51	0.5	0.51	0.51	0.48

$\mu = 0.52$
(276 predictions)

Script used:

```
#Author: Thomas Hollis
#Subject: Bachelor Thesis

#1. Data Import
data <- read.csv("R/BTC.csv")

#2. Data Split
data_train <- data[1:100,2]
data_test <- data[101:377,2]

#3. Model Train & apply Model
performance <- matrix(0,2,20)

for (j in 1:20)
{
  data_train <- data[1:100,2]
  c_count <- 0

  p <- round(runif(1, 1, 6))
  d <- 0
  q <- round(runif(1, 1, 6))

  for (i in 1:277)
  {
    btc_arima <- arima(data_train, order = c(p,d,q), method = "CSS")
    btc_pred <- predict(btc_arima, 1)

    if(i >= 2)
    {
      if(((btc_pred$pred[1] > data_test[i-1]) && (data_test[i] > data_test[i-1])) ||
((btc_pred$pred[1] < data_test[i-1]) && (data_test[i] < data_test[i-1])))
      {
        c_count = c_count+1
      }
    }
    data_train <- c(data_train, data_test[i])
  }
  performance[1,j] <- p*100+d*10+q
  performance[2,j] <- c_count/276.00
}
round(performance, digits = 2)
round(mean(performance[2,]), digits = 2)
```

4. ARIMA model predictions

Forex (EUR/GBP) accuracy:

556.00	532.00	433.00	364.00	522.0	521.00	434.00	233.00	334.00	455.00	254.00	444.0	656.0	243.00	144.00	625.00	141.00	425.00	452.00	441.00
0.51	0.54	0.52	0.54	0.5	0.51	0.51	0.51	0.51	0.52	0.51	0.5	0.5	0.51	0.51	0.51	0.49	0.53	0.52	0.52

$\mu = 0.51$
(234 predictions)

Cryptocurrency (BTC/USD) accuracy:

255.00	512.00	235.00	151.00	453.00	463.00	215.00	355.00	545.00	542.00	433.00	144.00	554.00	264.00	663.00	312.00	643.00	555.00	533.00	345.00
0.47	0.58	0.46	0.52	0.46	0.46	0.57	0.49	0.46	0.44	0.49	0.47	0.51	0.48	0.45	0.55	0.46	0.49	0.41	0.47

$\mu = 0.48$
(276 predictions)

Script used:

```
#Author: Thomas Hollis
#Subject: Bachelor Thesis

#1. Data Import
data <- read.csv("R/BTC.csv")

#2. Data Split
data_train <- data[1:100,2]
data_test <- data[101:377,2]

#3. Model Train & apply Model
performance <- matrix(0,2,20)

for (j in 1:20)
{
  data_train <- data[1:100,2]
  c_count <- 0

  p <- round(runif(1, 1, 6))
  d <- round(runif(1, 1, 6))
  q <- round(runif(1, 1, 6))

  for (i in 1:277)
  {
    btc_arima <- arima(data_train, order = c(p,d,q), method = "CSS")
    btc_pred <- predict(btc_arima, 1)

    if(i >= 2)
    {
      if(((btc_pred$pred[1] > data_test[i-1]) && (data_test[i] > data_test[i-1])) ||
((btc_pred$pred[1] < data_test[i-1]) && (data_test[i] < data_test[i-1])))
      {
        c_count = c_count+1
      }
    }
    data_train <- c(data_train, data_test[i])
  }
  performance[1,j] <- p*100+d*10+q
  performance[2,j] <- c_count/276.00
}
round(performance, digits = 2)
round(mean(performance[2,]), digits = 2)
```

5. ARCH model predictions

Forex (EUR/GBP) accuracy: ???% (??? predictions)

Cryptocurrency (BTC/USD) accuracy: ???% (??? predictions)

Script used:

```
#Author: Thomas Hollis
#Subject: Bachelor Thesis

#0. Package Import
library(fGarch)

#1. Data Import
data <- read.csv("R/BTC.csv")

#2. Data Split
data_train <- data[1:100,2]
data_test <- data[101:377,2]

#3. Model Train & apply Model
performance <- matrix(0,2,20)

for (j in 1:20)
{
  data_train <- data[1:100,2]
  c_count <- 0

  p <- 1
  q <- round(runif(1, 1, 6))

  for (i in 1:277)
  {
    btc_arch <- garchFit(~1+garch(p,q),data = data_train,trace=F)
    btc_pred <- predict(btc_arch, 1)

    if(i >= 2)
    {
      if(((btc_pred$pred[1] > data_test[i-1]) && (data_test[i] > data_test[i-1])) ||
((btc_pred$pred[1] < data_test[i-1]) && (data_test[i] < data_test[i-1])))
      {
        c_count = c_count+1
      }
    }

    data_train <- c(data_train, data_test[i])
  }
  performance[1,j] <- p*100+d*10+q
  performance[2,j] <- c_count/276.00
}
round(performance, digits = 2)
round(mean(performance[2,]), digits = 2)
```

6. GARCH model predictions

Forex (EUR/GBP) accuracy: ???% (??? predictions)

Cryptocurrency (BTC/USD) accuracy: ???% (??? predictions)

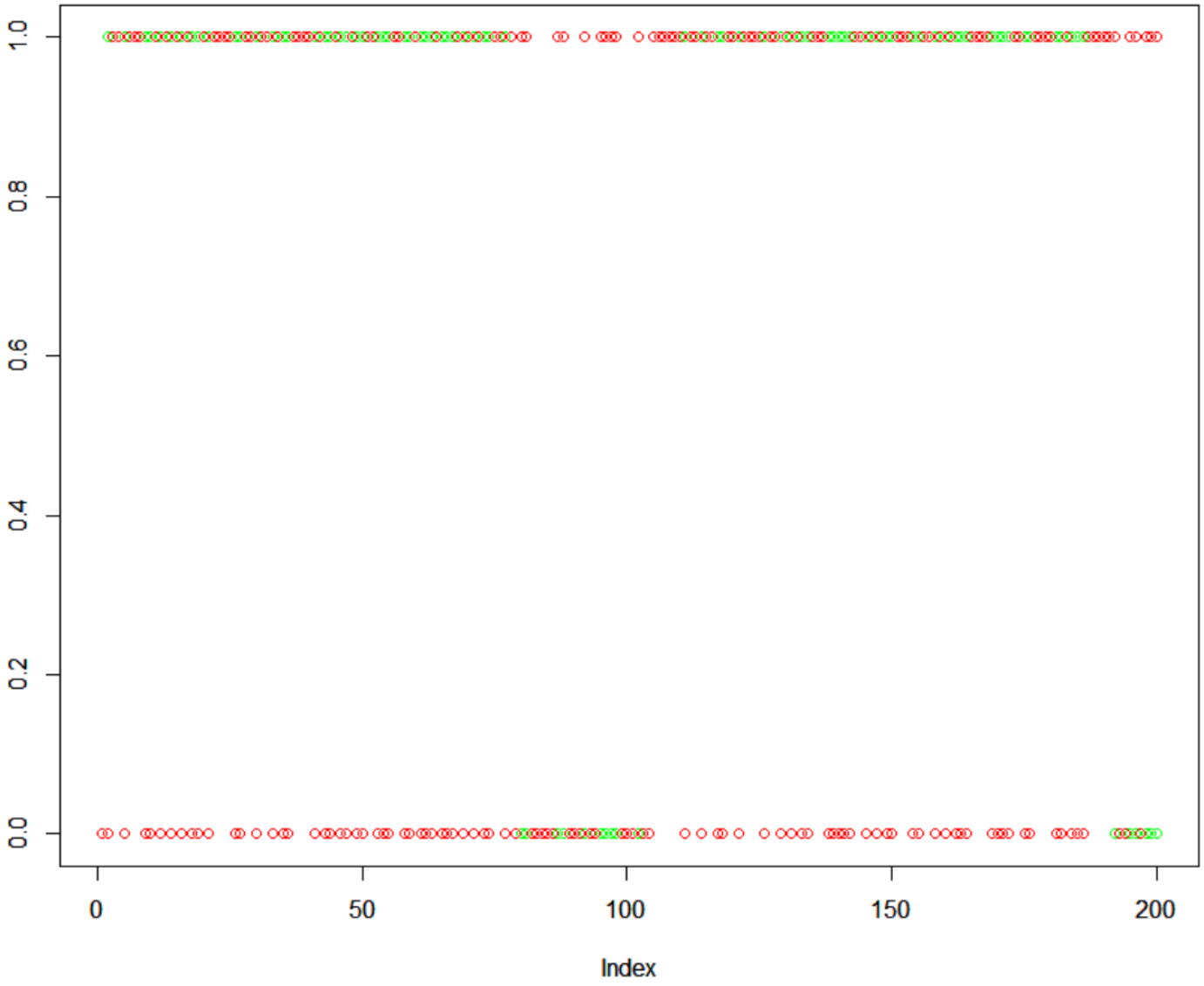
Script used:

```
#Author: Thomas Hollis
#Subject: Bachelor Thesis

#0. Package Import
library(fGarch)

#1. Data Import
data <- read.csv("R/BTC.csv")

#2. Data Split
data_train <- data[1:100,2]
data_test <- data[101:377,2]

#3. Model Train & apply Model
performance <- matrix(0,2,20)

for (j in 1:20)
{
  data_train <- data[1:100,2]
  c_count <- 0

  p <- round(runif(1, 1, 6))
  q <- round(runif(1, 1, 6))

  for (i in 1:277)
  {
    btc_arch <- garchFit(~1+garch(p,q),data = data_train,trace=F)
    btc_pred <- predict(btc_arch, 1)

    if(i >= 2)
    {
      if(((btc_pred$pred[1] > data_test[i-1]) && (data_test[i] > data_test[i-1])) ||
((btc_pred$pred[1] < data_test[i-1]) && (data_test[i] < data_test[i-1])))
      {
        c_count = c_count+1
      }
    }

    data_train <- c(data_train, data_test[i])
  }
  performance[1,j] <- p*100+d*10+q
  performance[2,j] <- c_count/276.00
}
round(performance, digits = 2)
round(mean(performance[2,]), digits = 2)
```

7. SLP model predictions

Forex (EUR/GBP) accuracy:
0.58 0.58 0.58 0.58 0.58 0.59 0.58 0.58 0.59 0.59 0.58 0.58 0.58 0.59 0.58 0.58 0.58 0.58 0.58 0.58
$\mu = 0.587$
(200 predictions)

Cryptocurrency (BTC/USD) accuracy:
0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52 0.52
$\mu = 0.518$
(200 predictions)



Script used:

```
#Author: Thomas Hollis
#Subject: Bachelor Thesis

#1. Data Import & Global Variables
data <- read.csv("R/DigitData2.csv") #import data

#2. Data post-processing
data_train <- data[1:300,2] #split data to training set
data_test <- data[301:501,2] #split data to testing set
acc = 0

#3. Model application
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

data_train <- normalize(data_train)
data_test <- normalize(data_test)

w <- runif(10, -0.2, 0.2)
b = 0
alpha = 0.1
y = integer(8)
right_value = 0

for(i in 1:290)
{
  trainingData <- data_train[i:(i+9)]

  if(data_train[i+10] > data_train[i+9])
  {
    right_value = 1
  }else
  {
    right_value = 0
  }

  if((trainingData%*%w + b) > 0)
  {
    y[i] = 1
  }else
  {
    y[i] = 0
  }

  w = w + alpha*(right_value - y[i])*right_value
  b = b + alpha*(right_value - y[i])
}


#prediction
real_result = integer(200)
result = integer(200)
```

```
for(i in 2:200)
{
  if(data_test[i] > data_test[i-1])
  {
    real_result[i] = 1
  }else
  {
    real_result[i] = 0
  }
}

for(i in 2:191)
{
  testData <- data_test[i:(i+9)]

  if((testData%*%w + b) > 0)
  {
    result[i] = 1
  }else
  {
    result[i] = 0
  }

  if(result[i] == real_result[i])
  {
    acc = acc+1
  }
}

plot(result, col = "green")
par(new = TRUE)
plot(real_result, col = "red")

result
real_result
acc/200
```

8. MLP model predictions

Forex (EUR/GBP) accuracy: ???% (??? predictions)

Cryptocurrency (BTC/USD) accuracy: ???% (??? predictions)

Script used:

```
(insert code here)
```

9. NN model predictions

Forex (EUR/GBP) accuracy: ???% (??? predictions)

Cryptocurrency (BTC/USD) accuracy: ???% (??? predictions)

Script used:

```
(insert code here)
```

10. SOM model predictions

Forex (EUR/GBP) accuracy: ???% (??? predictions)

Cryptocurrency (BTC/USD) accuracy: ???% (??? predictions)

Script used:

```
(insert code here)
```