ARIMA

Юрченков Иван Александрович

15 04 2022

Contents

0.1 Загрузка данных с yahoo.finance

```
library(quantmod)
```

```
##                 : xts
```

```
##                 : zoo
```

```
## 
##           : 'zoo'
```

```
##                   'package:base':
## 
##     as.Date, as.Date.numeric
```

```
##                 : TTR
```

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

```
quantmod::getSymbols(Symbols = "AAPL")
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
## 
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

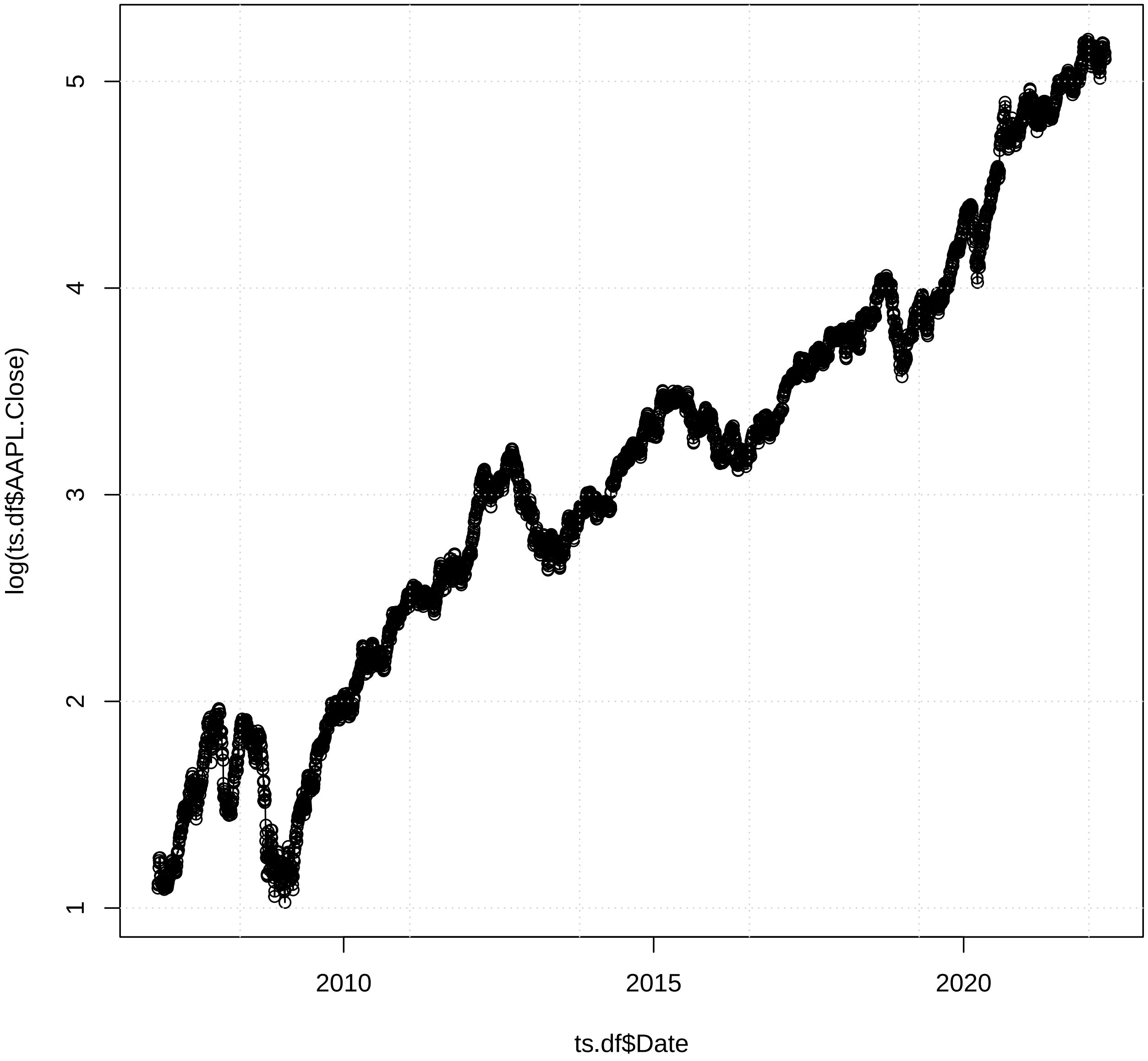
```
## [1] "AAPL"
```

```
ts.df <- data.frame(get("AAPL"))
ts.df$Date <- as.Date.character(rownames(ts.df), format = "%Y-%m-%d")
ts.df$Num_Date <- as.numeric(ts.df$Date)
head(ts.df)
```

```
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
## 2007-01-03  3.081786  3.092143 2.925000   2.992857  1238319600      2.562705
## 2007-01-04  3.001786  3.069643 2.993571   3.059286   847260400      2.619588
## 2007-01-05  3.063214  3.078571 3.014286   3.037500   834741600      2.600933
## 2007-01-08  3.070000  3.090357 3.045714   3.052500   797106800       2.613777
## 2007-01-09  3.087500  3.320714 3.041071   3.306071  3349298400       2.830903
## 2007-01-10  3.383929  3.492857 3.337500   3.464286  2952880000       2.966379
##                  Date Num_Date
## 2007-01-03 2007-01-03    13516
## 2007-01-04 2007-01-04    13517
## 2007-01-05 2007-01-05    13518
## 2007-01-08 2007-01-08    13521
## 2007-01-09 2007-01-09    13522
## 2007-01-10 2007-01-10    13523
```

$$x = \int_{-\infty}^{\infty} y \cdot f(x) dy$$

```
plot(x = ts.df$Date, y = log(ts.df$AAPL.Close), type = "o")
grid()
```

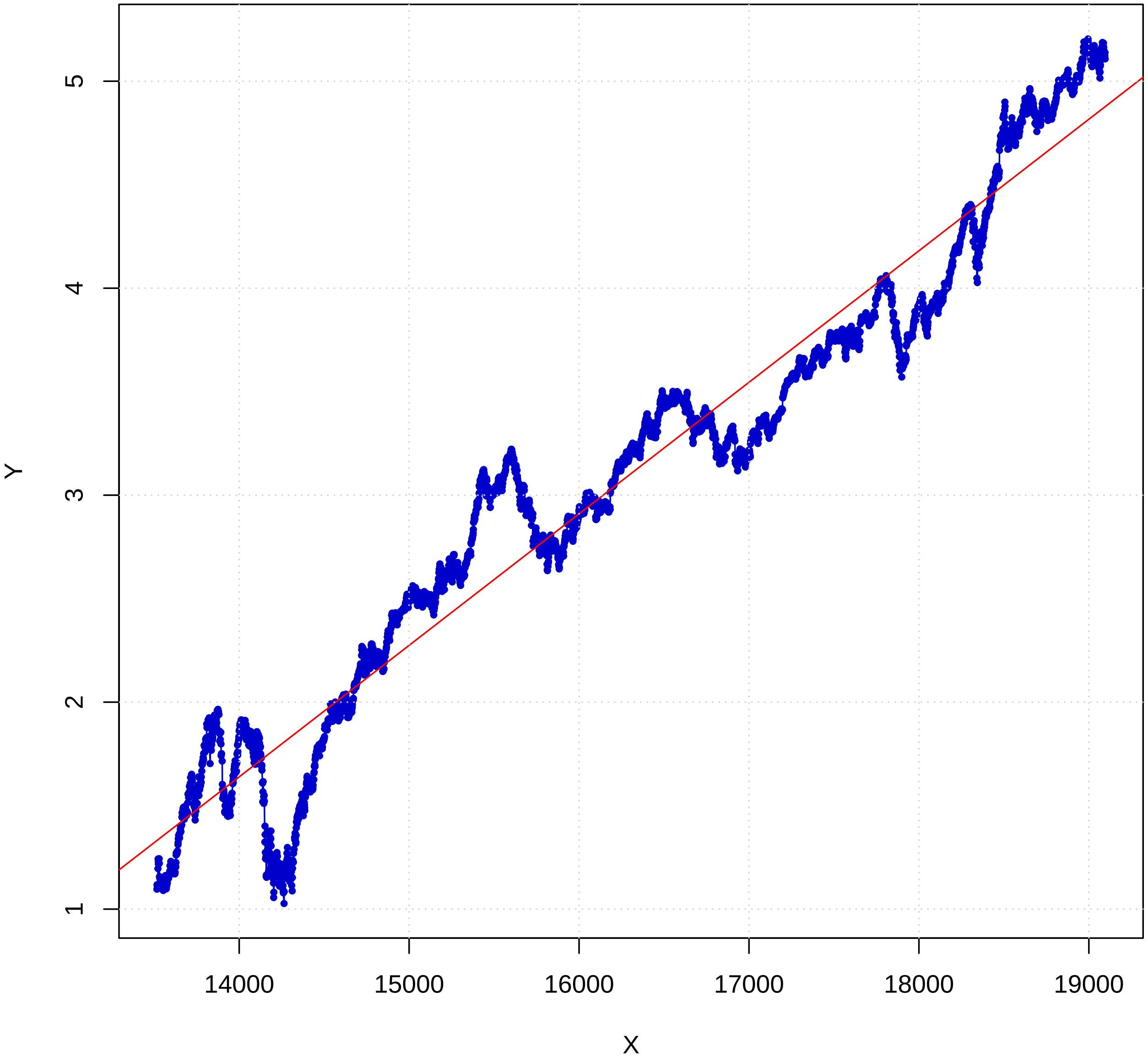


Вытаскиваю из графика линейный тренд

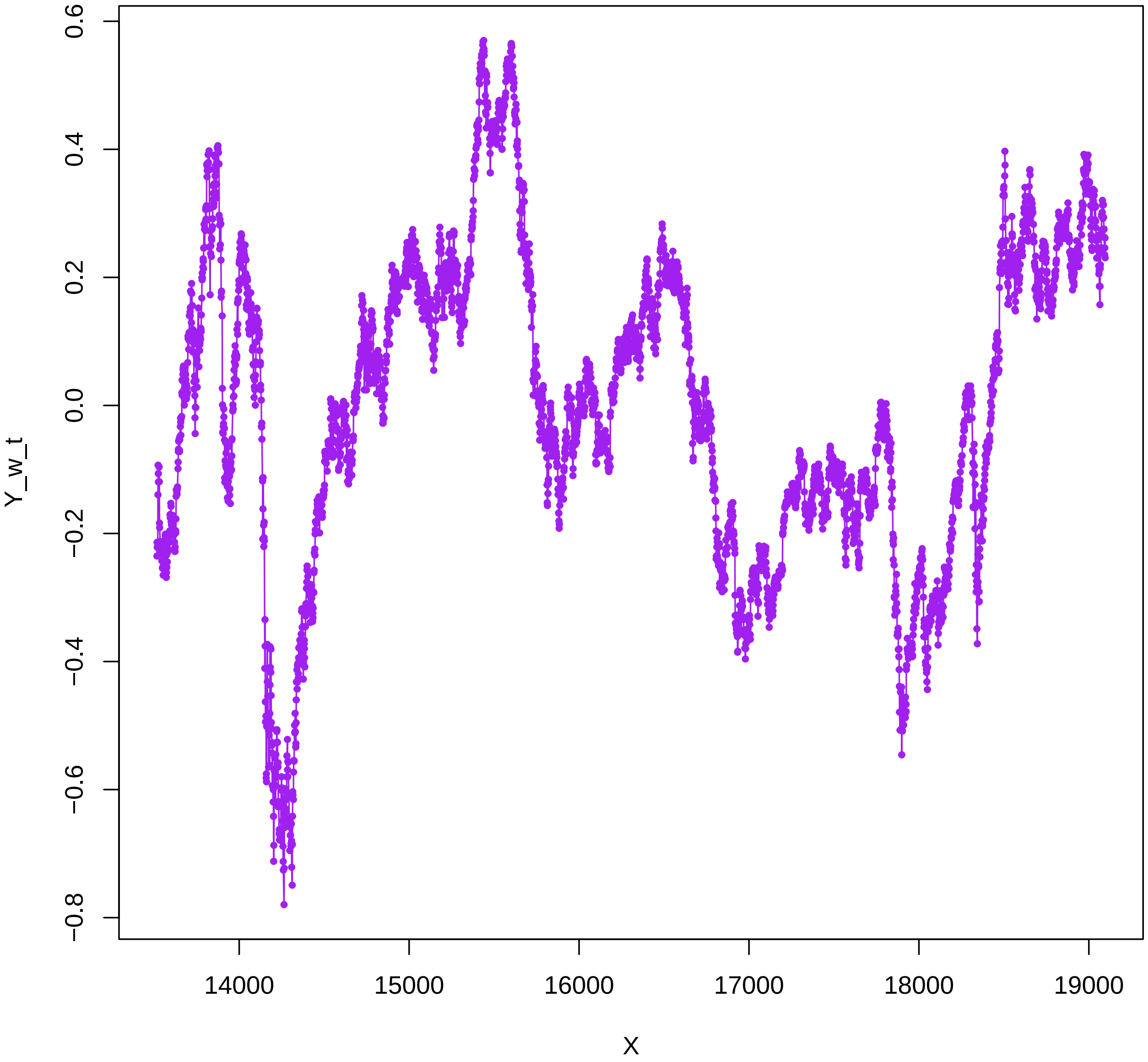
```
Y <- log(ts.df$AAPL.Close)
X <- ts.df$Num_Date
model_lm <- lm(Y ~ X)
model_lm
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Coefficients:
## (Intercept)            X
##  -7.2570214    0.0006354
```

```
plot(X, Y, type = "o", col = "blue3", pch = 19, cex = I(0.5))
abline(a = model_lm$coefficients[1], b = model_lm$coefficients[2], col ="red")
grid()
```



```
Y_w_t <- Y - model_lm$coefficients[1] - model_lm$coefficients[2] * X
plot(x = X, y = Y_w_t, col = "purple", type = "o", cex = I(0.5), pch = 19)
```



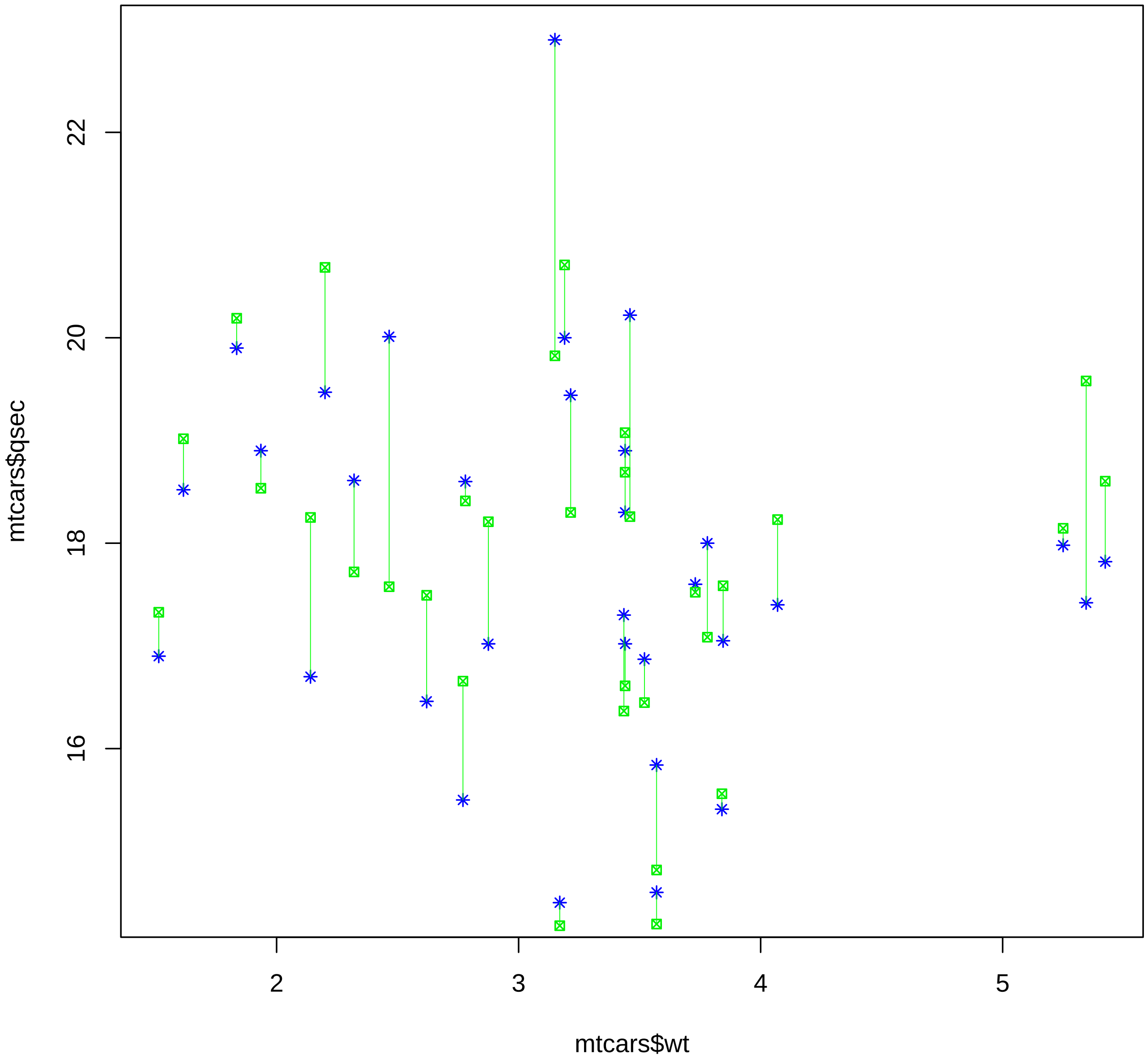
```
length(as.matrix(ts.df$AAPL.Close)[1:(1 + 10 + 1), 1])
```

```
## [1] 12
```

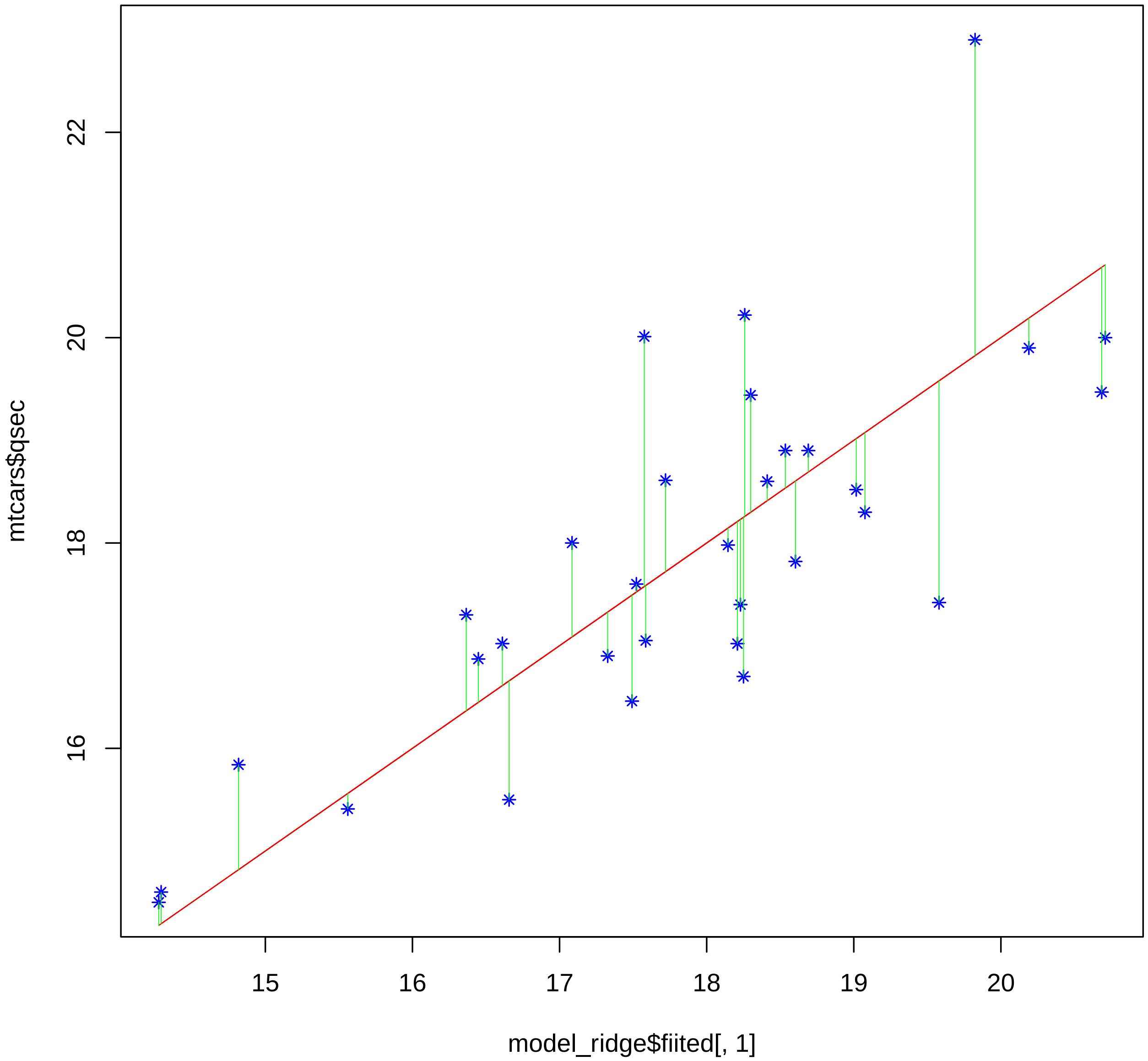
```
sliding_window <- function(numeric_vector, window_elements = 10) {
  rows_matrix <- length(numeric_vector) - window_elements + 1
  cols_matrix <- window_elements

  result_matrix <- matrix(nrow = rows_matrix, ncol = cols_matrix)

  for (index in 1:rows_matrix) {
    result_matrix[index, ] <- numeric_vector[index:(index - 1 + window_elements)]
  }

  return(result_matrix)
}
```

```r
lin_reg <- function(X, y) {
  X <- cbind(1, as.matrix(X))
  y <- as.matrix(y)
  w <- pracma::inv(t(X) %*% X) %*% t(X) %*% y
  fit <- X %*% w

  rownames(w)[1] <- "bias"
  return(list(parameters = w, fiited = fit))
}


lin_reg_ridge <- function(X, y, alpha = 0) {
  X <- cbind(1, as.matrix(X))
  y <- as.matrix(y)
  w <- pracma::inv(t(X) %*% X + alpha * diag(1, nrow = ncol(X))) %*% t(X) %*% y
  fit <- X %*% w

  rownames(w)[1] <- "bias"
  return(list(parameters = w, fiited = fit))
}
```

```r
model <- lin_reg(mtcars[, c(1, 3, 4)], mtcars[, c(5, 6)])
model_ridge <- lin_reg_ridge(mtcars[, c(1, 3, 4, 5, 6)], mtcars[, c(7)], alpha = 0.2)

plot(mtcars$wt, mtcars$qsec, col = "blue", cex = I(0.8), pch = 8)
points(mtcars$wt, model_ridge$fiited[, 1], col = "green2", cex = I(0.8), pch = 7)
segments(x0 = mtcars$wt, x1 = mtcars$wt, y0 = mtcars$qsec, y1 = model_ridge$fiited[, 1], col = "green",
```



```
plot(model_ridge$fiited[, 1], mtcars$qsec, col = "blue", cex = I(0.8), pch = 8)
lines(model_ridge$fiited[, 1], model_ridge$fiited[, 1], col = "red", lwd = I(0.6))
segments(x0 = model_ridge$fiited[, 1], x1 = model_ridge$fiited[, 1], y0 = mtcars$qsec, y1 = model_ridge$
```



```
cut_window <- function(data, history = 10, forward = 1) {
    rows <- nrow(data) - history - forward
    result_matrix <- matrix(nrow = rows, ncol = history + forward)
    for (index in 1:rows) {
        result_matrix[index, ] <- data[index:(index - 1 + history + forward), 1]
    }
    return(result_matrix)
}


his = 40
forwd = 5
data_X <- cut_window(data = as.matrix(ts.df$AAPL.Close), history = his, forward = forwd)
X <- cbind(1, data_X[,1:his])
Y <- data_X[,(his+1):(his + forwd)]

w = pracma::pinv(t(X) %*% X) %*% t(X) %*% Y
```

```
w
```

```
##               [,1]          [,2]         [,3]         [,4]          [,5]
##  [1,]  0.004733752  0.0103581346  0.018374788  0.025508615  0.0293432450
##  [2,]  0.047410917  0.0780981051  0.124656056  0.085129471  0.1691224124
##  [3,] -0.121224380 -0.1034509719 -0.120805637 -0.032415472 -0.1500155322
##  [4,]  0.069956206 -0.0519605553 -0.032062210 -0.051854848  0.0302025740
##  [5,] -0.059083511  0.0166504459 -0.096195643 -0.079183246 -0.0977328437
##  [6,]  0.049638582 -0.0153528197  0.050063894 -0.059564364 -0.0399145390
##  [7,]  0.005618950  0.0523991689 -0.017594916  0.050580830 -0.0612582440
##  [8,] -0.008398499 -0.0002975257  0.050624625 -0.021607425  0.0488602401
##  [9,] -0.018040067 -0.0294054276 -0.026873282  0.028310648 -0.0485794190
## [10,]  0.029492870  0.0127430341  0.003427688  0.002539006  0.0659764730
## [11,]  0.029654719  0.0617468181  0.052738341  0.039960726  0.0296347544
## [12,]  0.003481259  0.0334322870  0.066073307  0.056530401  0.0445439817
## [13,]  0.019036018  0.0185881326  0.041578980  0.077230855  0.0673511479
## [14,] -0.087314174 -0.0679161784 -0.070177234 -0.042887983 -0.0096085230
## [15,]  0.079975161 -0.0028704326  0.026861497  0.015532127  0.0489089881
## [16,] -0.010422300  0.0655390920 -0.025109326  0.009365425 -0.0040940470
## [17,]  0.008562191 -0.0030324156  0.070386997 -0.019736592  0.0177595127
## [18,] -0.039553669 -0.0315118597 -0.045117693  0.030998941 -0.0610145731
## [19,]  0.020723483 -0.0156154930 -0.001608569 -0.019939842  0.0618592966
## [20,] -0.024435594 -0.0027521210 -0.035219629 -0.019773005 -0.0539630576
## [21,]  0.096106917  0.0661922281  0.078738060  0.046174067  0.0710574076
## [22,] -0.115754005 -0.0144155032 -0.039804568 -0.027315727 -0.0498417629
## [23,] -0.065804871 -0.1836089846 -0.087117420 -0.106494839 -0.1016845836
## [24,]  0.157281711  0.0955228216 -0.008429023  0.076158003  0.0530599030
## [25,] -0.080474915  0.0787121315  0.018290845 -0.083312330 -0.0005625805
## [26,]  0.021555228 -0.0616870484  0.092037546  0.032977575 -0.0641766889
## [27,] -0.033598175 -0.0103225825 -0.091077497  0.062745872  0.0022094732
## [28,]  0.061249812  0.0273873964  0.051880192 -0.031888167  0.1237442558
## [29,] -0.091939876 -0.0312806950 -0.068229490 -0.037941912 -0.1281962487
## [30,]  0.040220126 -0.0511737752  0.010650620 -0.029429354  0.0079002877
## [31,]  0.026430668  0.0669929948 -0.022592128  0.037629514 -0.0038593893
## [32,] -0.024239196  0.0007225409  0.037381602 -0.049563102  0.0118928546
## [33,]  0.070267981  0.0437086850  0.067110516  0.102745110  0.0124021925
## [34,] -0.100663997 -0.0275052489 -0.051347547 -0.025499252  0.0075447661
## [35,]  0.080916867 -0.0234746234  0.044406310  0.019866481  0.0498214250
## [36,] -0.057468499  0.0222430149 -0.086806614 -0.014900066 -0.0373916086
## [37,]  0.024728066 -0.0304757755  0.055127267 -0.057219169  0.0110883407
## [38,]  0.038461635  0.0624192922  0.007838016  0.092580423 -0.0244423061
## [39,] -0.029726943  0.0069476828  0.024819897 -0.027075523  0.0666027092
## [40,]  0.009238145 -0.0221012902  0.012930358  0.032437496 -0.0261183519
## [41,]  0.979295149  0.9724471741  0.941875392  0.942449339  0.9664831624
```