5_Nile_HoltWinters.R

felixreichel

2021-10-26

```
# Course: Time series analysis
# Exercise: 5th / Nile Holt-Winters
# Author: Felix Reichel
```

```
require(astsa)
```

```
## Loading required package: astsa
```

```
require(tseries)
```

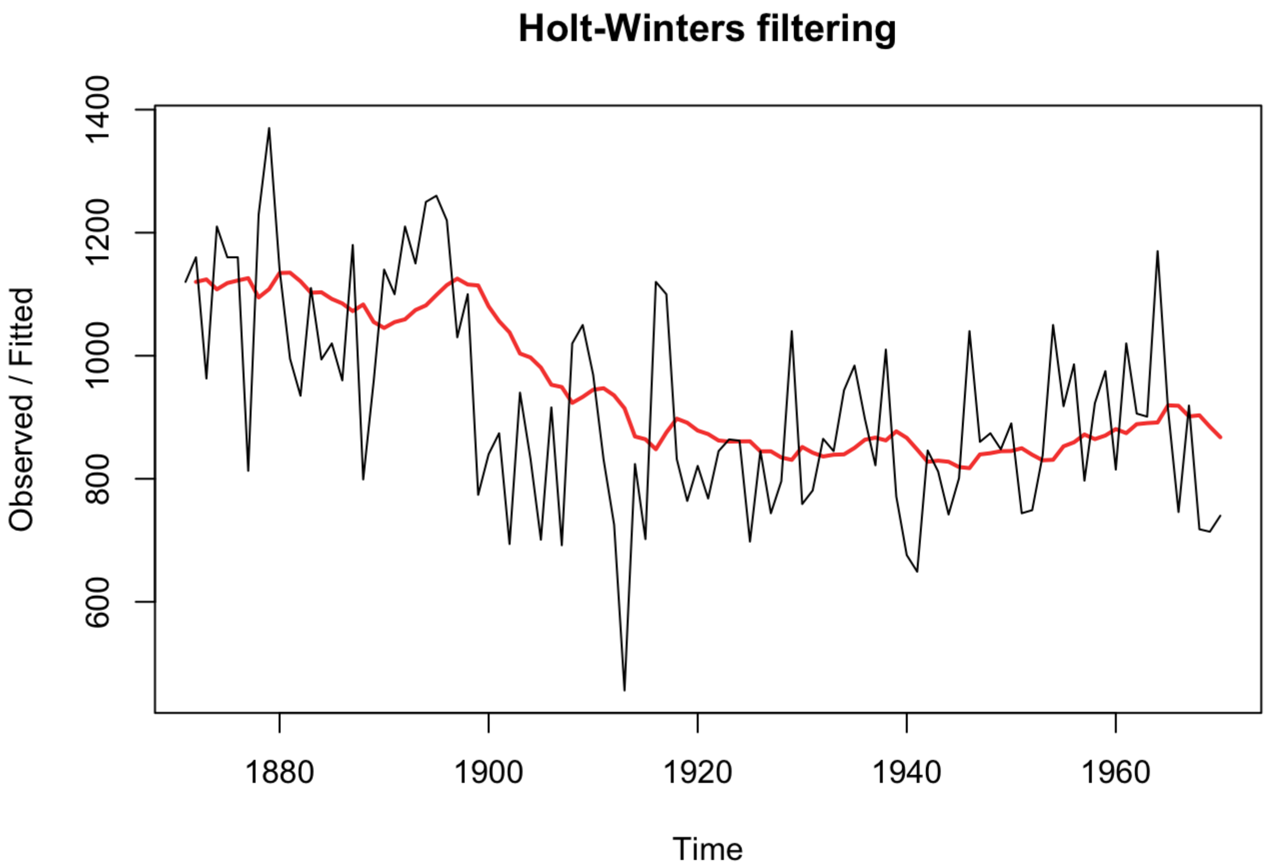
```
## Loading required package: tseries
```

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

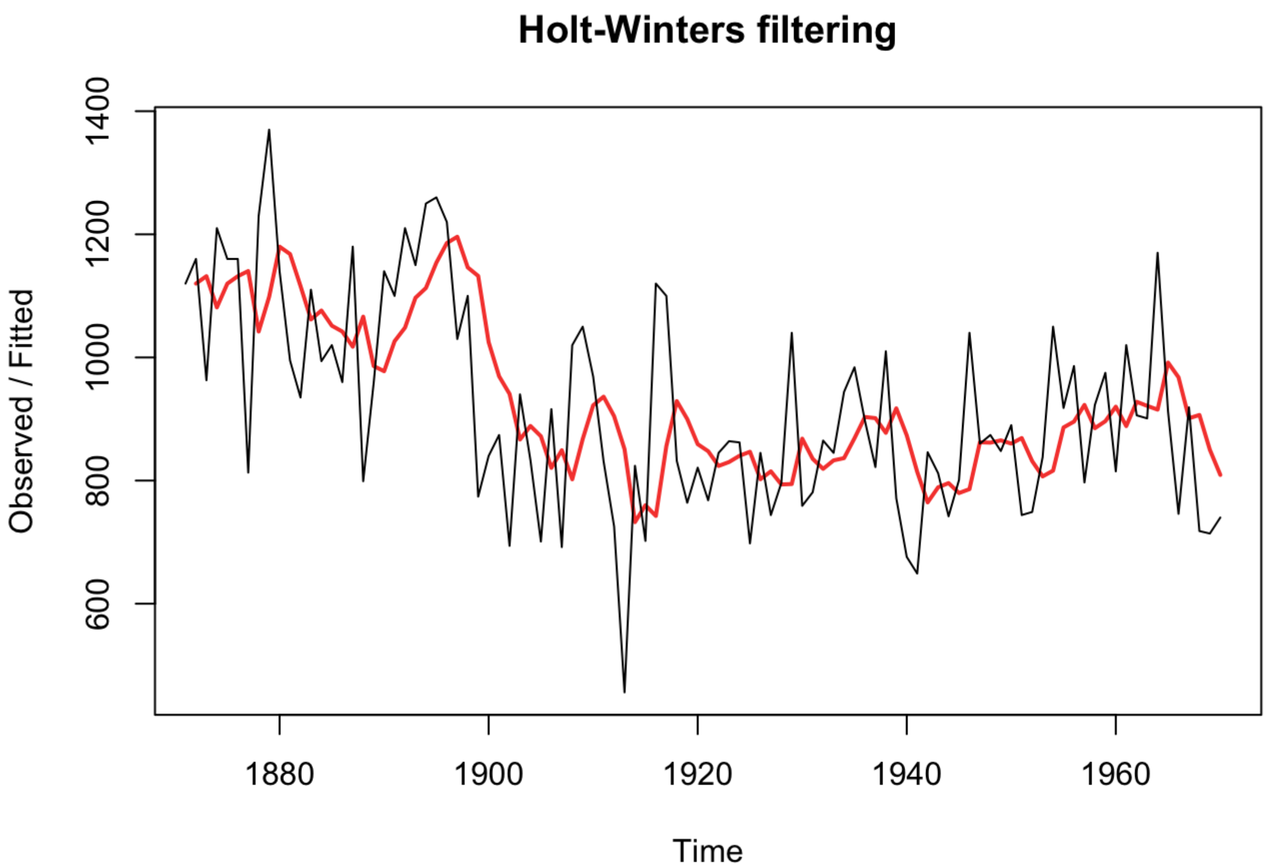
```
require(Metrics)
```

```
## Loading required package: Metrics
```

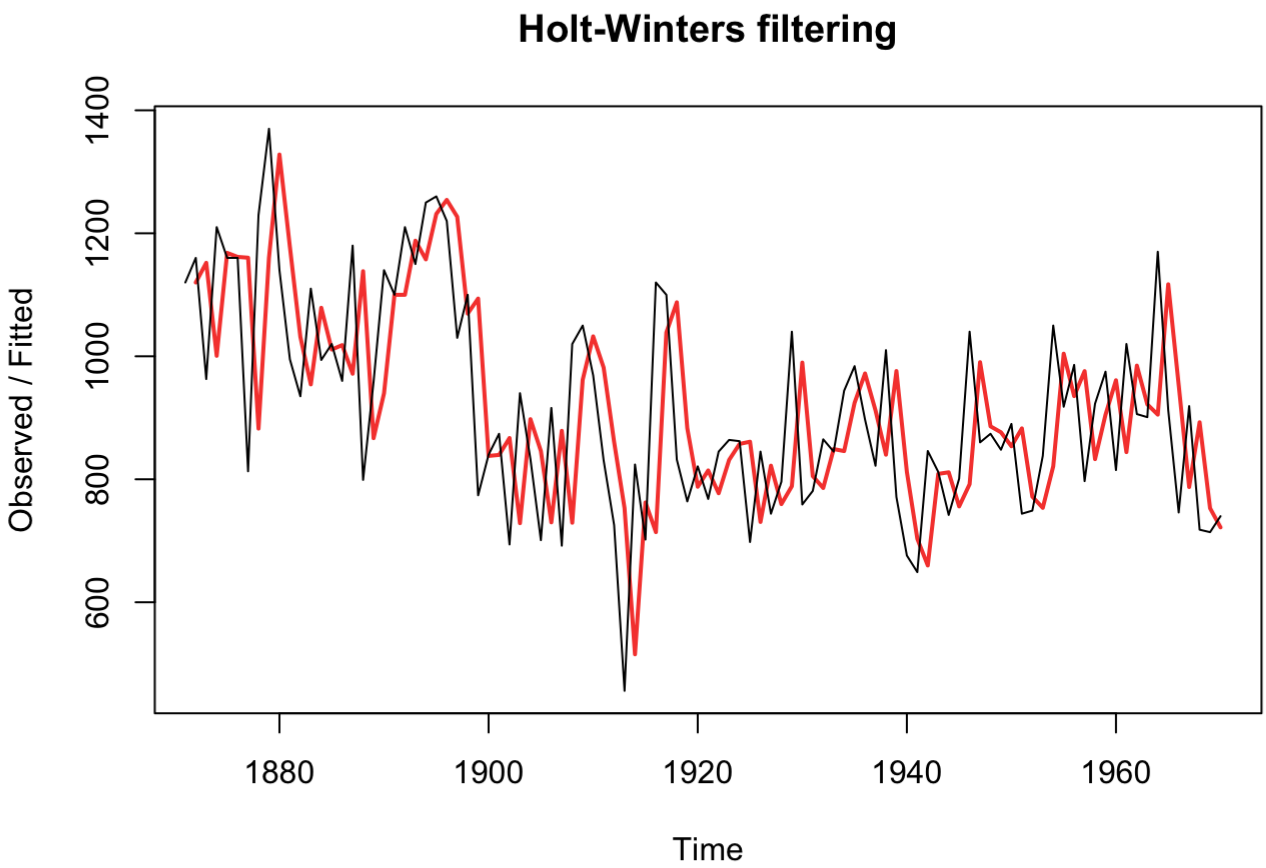
```
# 1.)
alpha_1 = 0.1
Nile_exp_1 = HoltWinters(Nile, alpha = alpha_1, beta = FALSE, gamma = FALSE)

alpha_2 = 0.3
Nile_exp_2 = HoltWinters(Nile, alpha = alpha_2, beta = FALSE, gamma = FALSE)

alpha_3 = 0.8
Nile_exp_3 = HoltWinters(Nile, alpha = alpha_3, beta = FALSE, gamma = FALSE)

plot(Nile_exp_1, lwd = 2)
```




```
plot(Nile_exp_2, lwd = 2)
```



```
plot(Nile_exp_3, lwd = 2)
```




```
# 2.)
actual_values <- c(Nile[2:100])
actual_values
```

```
##  [1] 1160  963 1210 1160 1160  813 1230 1370 1140  995  935 1110  994 1020  960
## [16] 1180  799  958 1140 1100 1210 1150 1250 1260 1220 1030 1100  774  840  874
## [31]  694  940  833  701  916  692 1020 1050  969  831  726  456  824  702 1120
## [46] 1100  832  764  821  768  845  864  862  698  845  744  796 1040  759  781
## [61]  865  845  944  984  897  822 1010  771  676  649  846  812  742  801 1040
## [76]  860  874  848  890  744  749  838 1050  918  986  797  923  975  815 1020
## [91]  906  901 1170  912  746  919  718  714  740
```

```
predicted_values_1 <- c(Nile_exp_1$fitted[1:99])
predicted_values_1
```

```
##  [1] 1120.0000 1124.0000 1107.9000 1118.1100 1122.2990 1126.0691 1094.7622
##  [8] 1108.2860 1134.4574 1135.0116 1121.0105 1102.4094 1103.1685 1092.2516
## [15] 1085.0265 1072.5238 1083.2714 1054.8443 1045.1599 1054.6439 1059.1795
## [22] 1074.2615 1081.8354 1098.6519 1114.7867 1125.3080 1115.7772 1114.1995
## [29] 1080.1795 1056.1616 1037.9454 1003.5509  997.1958  980.7762  952.7986
## [36]  949.1187  923.4069  933.0662  944.7596  947.1836  935.5652  914.6087
## [43]  868.7478  864.2731  848.0458  875.2412  897.7171  891.1454  878.4308
## [50]  872.6877  862.2190  860.4971  860.8474  860.9626  844.6664  844.6997
## [57]  834.6298  830.7668  851.6901  842.4211  836.2790  839.1511  839.7360
## [64]  850.1624  863.5461  866.8915  862.4024  877.1621  866.5459  847.4913
## [71]  827.6422  829.4780  827.7302  819.1572  817.3414  839.6073  841.6466
## [78]  844.8819  845.1937  849.6744  839.1069  830.0962  830.8866  852.7979
## [85]  859.3181  871.9863  864.4877  870.3389  880.8050  874.2245  888.8021
## [92]  890.5219  891.5697  919.4127  918.6714  901.4043  903.1639  884.6475
## [99]  867.5827
```

```
mse1 <- mse(actual_values, predicted_values_1)
mse1
```

```
## [1] 21495.81
```

```
mae1 <- mae(actual_values, predicted_values_1)
mae1
```

```
## [1] 114.3894
```

```
mape1 <- mape(actual_values, predicted_values_1)
mape1
```

```
## [1] 0.1348183
```

```
predicted_values_2 <- c(Nile_exp_2$fitted[0:99])
predicted_values_2
```

```
##  [1] 1120.0000 1132.0000 1081.3000 1119.9100 1131.9370 1140.3559 1042.1491
##  [8] 1098.5044 1179.9531 1167.9672 1116.0770 1061.7539 1076.2277 1051.5594
## [15] 1042.0916 1017.4641 1066.2249  986.0574  977.6402 1026.3481 1048.4437
## [22] 1096.9106 1112.8374 1153.9862 1185.7903 1196.0532 1146.2373 1132.3661
## [29] 1024.8563  969.3994  940.7796  866.7457  888.7220  872.0054  820.7038
## [36]  849.2926  802.1048  867.4734  922.2314  936.2620  904.6834  851.0784
## [43]  732.5549  759.9884  742.5919  855.8143  929.0700  899.9490  859.1643
## [50]  847.7150  823.8005  830.1604  840.3123  846.8186  802.1730  815.0211
## [57]  793.7148  794.4003  868.0802  835.3562  819.0493  832.8345  836.4842
## [64]  868.7389  903.3172  901.4221  877.5954  917.3168  873.4218  814.1952
## [71]  764.6367  789.0457  795.9320  779.7524  786.1267  862.2887  861.6021
## [78]  865.3214  860.1250  869.0875  831.5613  806.7929  816.1550  886.3085
## [85]  895.8160  922.8712  885.1098  896.4769  920.0338  888.5237  927.9666
## [92]  921.3766  915.2636  991.6845  967.7792  901.2454  906.5718  850.0003
## [99]  809.2002
```

```
mse2 <- mse(actual_values, predicted_values_2)
mse2
```

```
## [1] 20637.51
```

```
mae2 <- mae(actual_values, predicted_values_2)
mae2
```

```
## [1] 113.6598
```

```
mape2 <- mape(actual_values, predicted_values_2)
mape2
```

```
## [1] 0.1308681
```

```
predicted_values_3 <- c(Nile_exp_3$fitted[0:99])
predicted_values_3
```

```
##  [1] 1120.0000 1152.0000 1000.8000 1168.1600 1161.6320 1160.3264  882.4653
##  [8] 1160.4931 1328.0986 1177.6197 1031.5239  954.3048 1078.8610 1010.9722
## [15] 1018.1944  971.6389 1138.3278  866.8656  939.7731 1099.9546 1099.9909
## [22] 1187.9982 1157.5996 1231.5199 1254.3040 1226.8608 1069.3722 1093.8744
## [29]  837.9749  839.5950  867.1190  728.6238  897.7248  845.9450  729.9890
## [36]  878.7978  729.3596  961.8719 1032.3744  981.6749  861.1350  753.0270
## [43]  515.4054  762.2811  714.0562 1038.8112 1087.7622  883.1524  787.8305
## [50]  814.3661  777.2732  831.4546  857.4909  861.0982  730.6196  822.1239
## [57]  759.6248  788.7250  989.7450  805.1490  785.8298  849.1660  845.8332
## [64]  924.3666  972.0733  912.0147  840.0029  976.0006  812.0001  703.2000
## [71]  659.8400  808.7680  811.3536  755.8707  791.9741  990.3948  886.0790
## [78]  876.4158  853.6832  882.7366  771.7473  753.5495  821.1099 1004.2220
## [85]  935.2444  975.8489  832.7698  904.9540  960.9908  844.1982  984.8396
## [92]  921.7679  905.1536 1117.0307  953.0061  787.4012  892.6802  752.9360
## [99]  721.7872
```

```
mse3 <- mse(actual_values, predicted_values_3)
mse3
```

```
## [1] 24386.14
```

```
mae3 <- mae(actual_values, predicted_values_3)
mae3
```

```
## [1] 124.0603
```

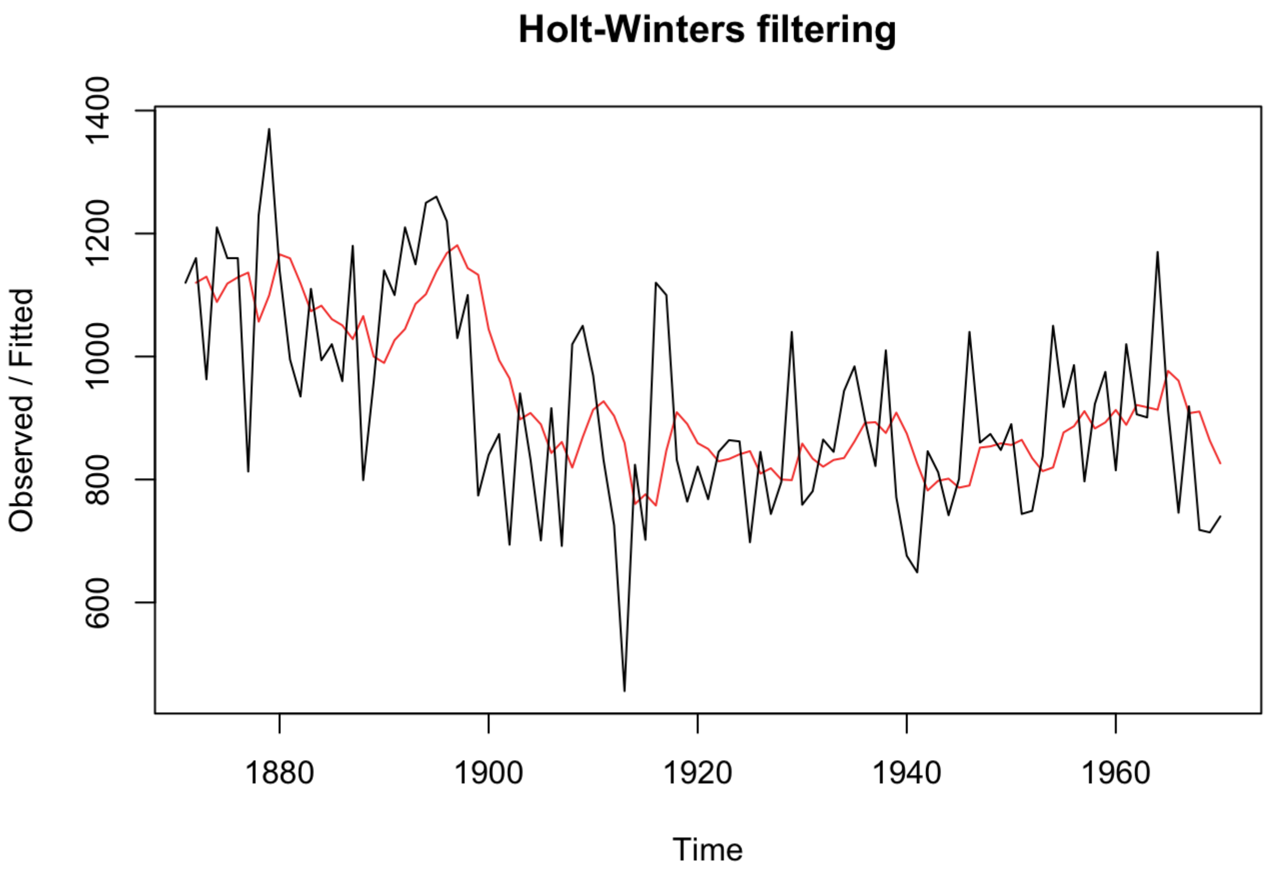
```
mape3 <- mape(actual_values, predicted_values_3)
mape3
```

```
## [1] 0.1405985
```

```
# in my chosen grid: alpha = 0.3

# 3.)
# -> Unknown parameters are determined by minimizing the squared one-step prediction
 error.
Nile_exp_opt = HoltWinters(Nile, beta = FALSE, gamma = FALSE)
alpha_opt <- Nile_exp_opt$alpha
alpha_opt
```

```
## [1] 0.2465579
```

```
plot(Nile_exp_opt)
```



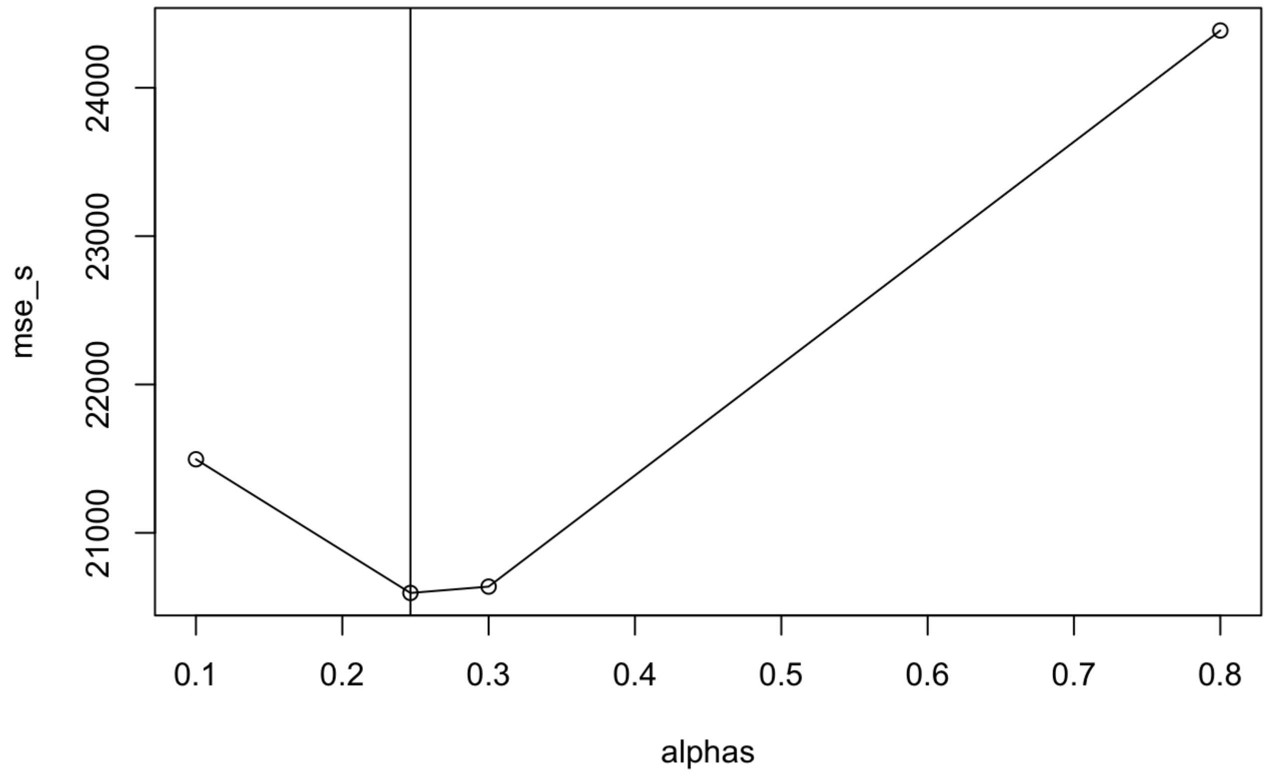

```
predicted_values_opt <- c(Nile_exp_opt$fitted[0:99])
predicted_values_opt
```

```
##  [1] 1120.0000 1129.8623 1088.7211 1118.6234 1128.8251 1136.5115 1056.7472
##  [8] 1099.4640 1166.1668 1159.7152 1119.1034 1073.7112 1082.6585 1060.7991
## [15] 1050.7397 1028.3671 1065.7534  999.9833  989.6320 1026.7064 1044.7775
## [22] 1085.5144 1101.4138 1138.0489 1168.1169 1180.9091 1143.7013 1132.9264
## [29] 1044.4303  994.0264  964.4329  897.7556  908.1713  889.6372  843.1272
## [36]  861.0946  819.4030  868.8617  913.5228  927.2011  903.4820  859.7224
## [43]  760.1815  775.9164  757.6918  847.0217  909.3955  890.3130  859.1696
## [50]  849.7586  829.6003  833.3972  840.9426  846.1345  809.6107  818.3362
## [57]  800.0081  799.0198  858.4354  833.9188  820.8713  831.7516  835.0181
## [64]  861.8884  891.9960  893.2298  875.6675  908.7882  874.8155  825.7959
## [71]  782.2055  797.9345  801.4025  786.7563  790.2682  851.8416  853.8531
## [78]  858.8205  856.1526  864.4979  834.7882  813.6365  819.6435  876.4397
## [85]  886.6867  911.1732  883.0229  892.8796  913.1270  888.9330  921.2486
## [92]  917.4889  913.4235  976.6844  960.7360  907.7911  910.5548  863.0789
## [99]  826.3223
```

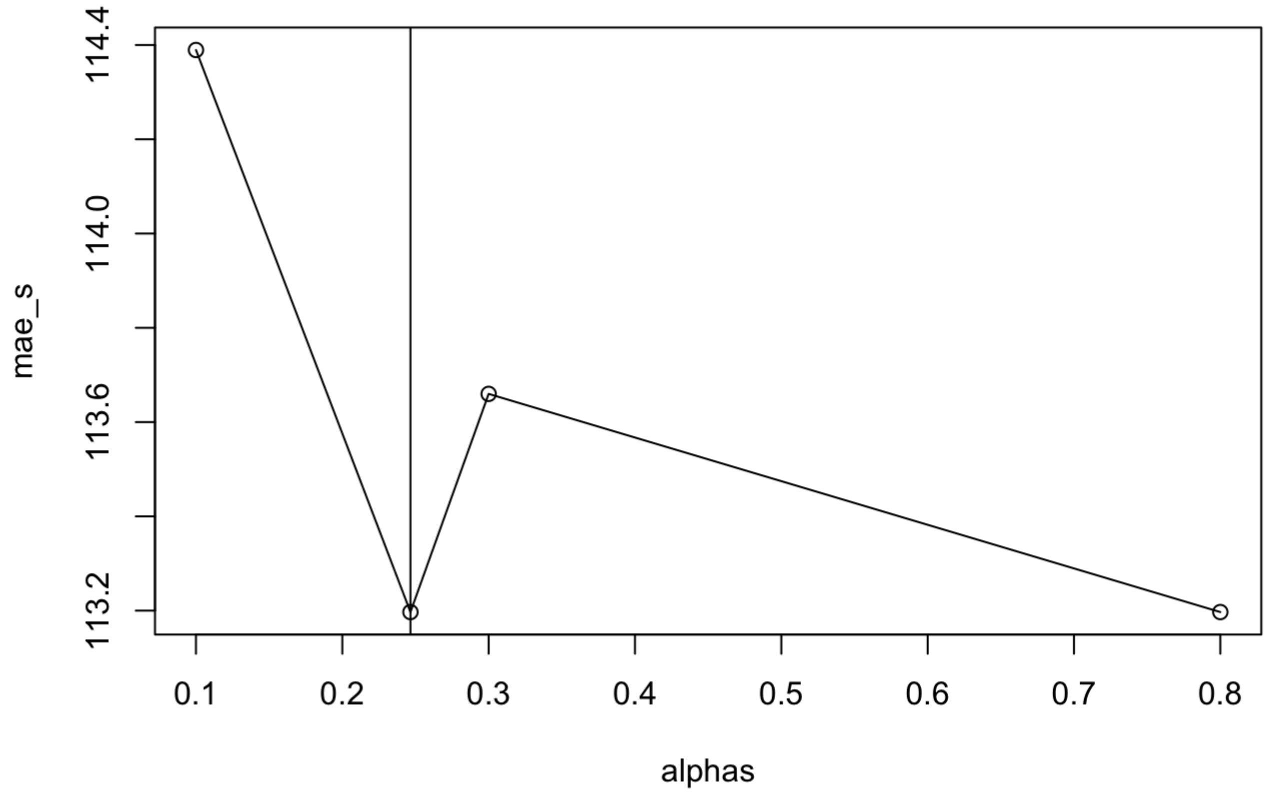
```
mse4 <- mse(actual_values, predicted_values_opt)
mae4 <- mae(actual_values, predicted_values_opt)
mape4 <- mape(actual_values, predicted_values_opt)
mape4 # ~ 13.07% mean abs. perc. err.
```

```
## [1] 0.1307089
```

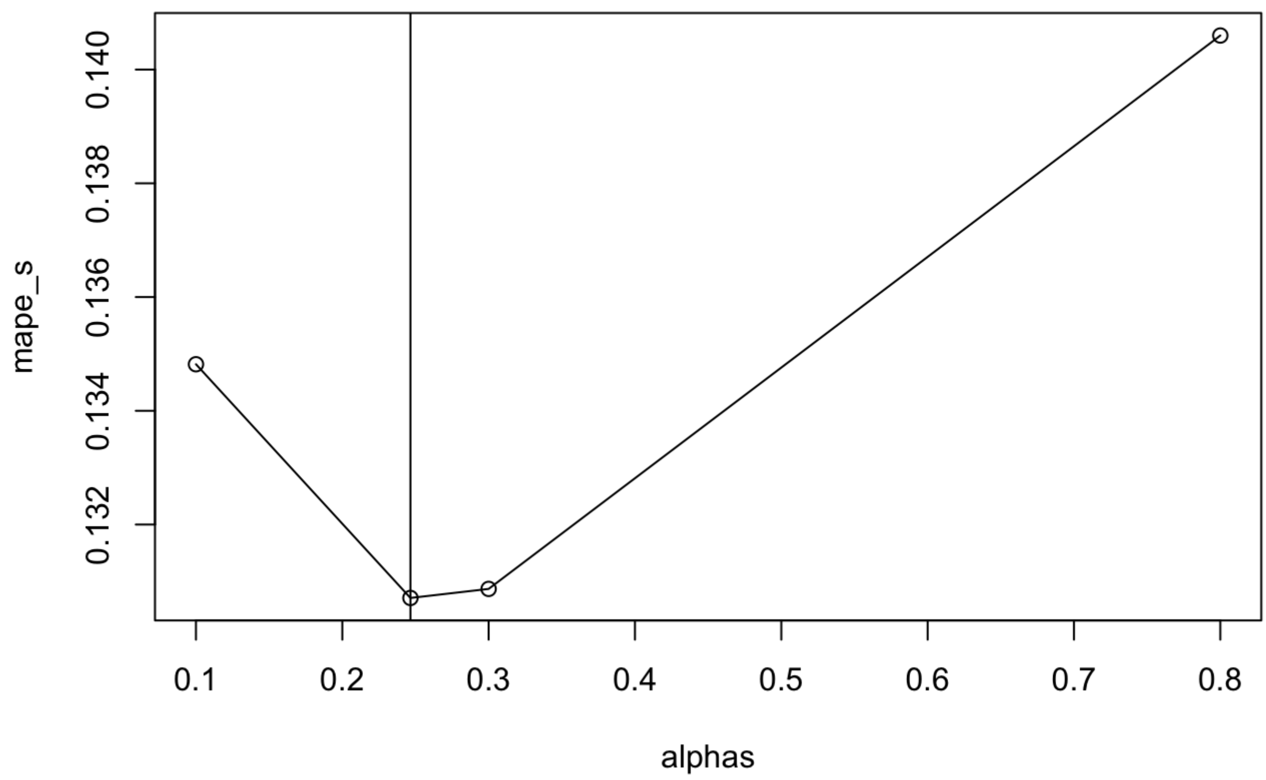
```
# 4.)
alphas <- c(alpha_1, alpha_opt, alpha_2, alpha_3)

mse_s <- c(mse1, mse4, mse2, mse3)
mae_s <- c(mae1, mae4, mae2, mae4)
mape_s <- c(mape1, mape4, mape2, mape3)

# MSE
df_mse <- data.frame(alphas, mse_s)
# could fit a function using a linear model
plot(df_mse)
lines(df_mse)
abline(v = alpha_opt)
```




```
# MAE
df_mae <- data.frame(alphas, mae_s)
plot(df_mae)
lines(df_mae)
abline(v = alpha_opt)
```




```
# MAPE
df_mape <- data.frame(alphas, mape_s)
plot(df_mape)
lines(df_mape)
abline(v = alpha_opt)
```



mape_s
0.132
0.134
0.136
0.138
0.140
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
alphas