Results

library(dplyr)  
library(knitr)  
library(ggplot2)  
library(RColorBrewer)  
library(smoothCV)  
library(data.table)  
library(forecast)  
library(TTR)

## Data Preparation

Set start date to 1 January 2017 and set end date to current system date. Take crude oil price data in the specified period from Yahoo finance:

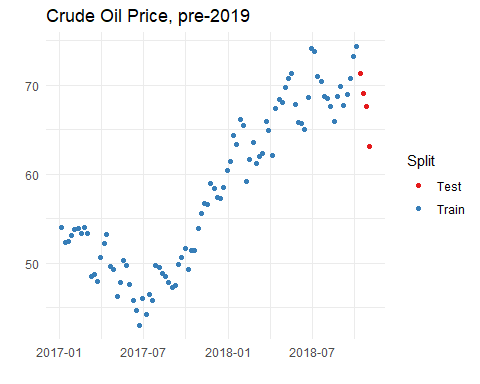
start <- as.POSIXct("2017-01-01")  
end <- as.POSIXct(Sys.Date())  
  
crude\_wk <- tidyquant::tq\_get("CL=F",   
 from = start, to = end) |>  
 mutate(date = as.Date(date, format ="%d/%m/%Y"),  
 week = ISOweek::ISOweek(date))|>  
 group\_by(week) |> slice\_tail(n = 1) |>  
 select(date, week, close) |> ungroup()  
  
crude\_wk |> head() |>   
 kable(caption = "First five observations in the dataset")

First five observations in the dataset

| date | week | close |
| --- | --- | --- |
| 2017-01-06 | 2017-W01 | 53.99 |
| 2017-01-13 | 2017-W02 | 52.37 |
| 2017-01-20 | 2017-W03 | 52.42 |
| 2017-01-27 | 2017-W04 | 53.17 |
| 2017-02-03 | 2017-W05 | 53.83 |
| 2017-02-10 | 2017-W06 | 53.86 |

## Case the First: pre-2019

trainset <- crude\_wk |>   
 filter(date <= as.Date("2018-10-10")) |>  
 mutate(Split = "Train")  
testset <- crude\_wk |>   
 filter(date > as.Date("2018-10-10")) |>  
 head(4) |> mutate(Split = "Test")  
  
bind\_rows(trainset,testset) |>   
 ggplot(aes(x = date, y = close,   
 color = Split)) +  
 geom\_point() +  
 scale\_color\_brewer(palette="Set1") +  
 theme\_minimal() +  
 xlab ("") + ylab ("") +  
 ggtitle("Crude Oil Price, pre-2019")



We pick a point where the weak positive trend previously established transitions into a negative trend.

### Aggregation

#### DES

Now we optimize parameters in the training set through cross-validation, and generate an aggregation table for DES:

resultsDES <- trainset |> select(close) |>  
 as.data.table() |>   
 fcCV(fullset = \_,   
 initialn=12,   
 folds = 20,   
 "DES",   
 alphrange=seq(0.1,1,0.1),   
 betarange=seq(0.1,1,0.1))  
  
aggregateDES <- resultsDES[[3]] |>  
 as\_tibble() |>  
 group\_by(alphrange, betarange) |>  
 summarise(mean\_MSE=mean(MSE),  
 var\_MSE=var(MSE),  
 mean\_MAPE=mean(MAPE),  
 var\_MAPE=var(MAPE),  
 mean\_MAE=mean(MAE),  
 var\_MAE=var(MAE)) |>  
 ungroup()  
  
kable(head(aggregateDES),   
 col.names = c("$\\alpha$", "$\\beta$",  
 "mean (MSE)", "variance (MSE)",  
 "mean (MAPE)", "variance (MAPE)",  
 "mean (MAE)", "variance (MAE)"))

|  |  | mean (MSE) | variance (MSE) | mean (MAPE) | variance (MAPE) | mean (MAE) | variance (MAE) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0.1 | 0.1 | 24.21855 | 566.60778 | 7.628404 | 19.576321 | 4.257092 | 4.616110 |
| 0.1 | 0.2 | 13.24635 | 184.65251 | 5.132368 | 9.115279 | 2.976714 | 2.631998 |
| 0.1 | 0.3 | 13.03078 | 94.16509 | 5.310935 | 5.911326 | 3.063481 | 1.752101 |
| 0.1 | 0.4 | 13.55964 | 61.36210 | 5.510104 | 3.354422 | 3.185610 | 1.066899 |
| 0.1 | 0.5 | 13.62270 | 97.33306 | 5.298023 | 5.186206 | 3.081098 | 1.671368 |
| 0.1 | 0.6 | 13.83147 | 175.04999 | 5.120507 | 7.628158 | 3.002294 | 2.520726 |

We find parameters that optimize each measure:

params <- aggregateDES |>  
 select(mean\_MSE, mean\_MAPE, mean\_MAE,  
 var\_MSE, var\_MAPE, var\_MAE)   
  
bestparams <- params |>   
 apply(FUN = slice\_min, MARGIN = 2,   
 .data = aggregateDES, n = 2) |>  
 purrr::map2(names(params), select,   
 alphrange, betarange)  
  
titles <- names(bestparams) |>   
 stringr::str\_split(pattern="\_")  
  
for(p in seq(length(bestparams))){  
 paste("Parameters that minimize",   
 titles[[p]][1], "of", titles[[p]][2]) |>  
 kable(bestparams[[p]],   
 caption = \_,  
 col.names = c(  
 paste(  
 {titles[[p]][1] |>   
 R.utils::capitalize()},   
 "of", titles[[p]][2]),  
 "$\\alpha$", "$\\beta$"))|> print(p)  
}

Parameters that minimize mean of MSE

| Mean of MSE |  |  |
| --- | --- | --- |
| 11.53505 | 0.2 | 0.2 |
| 12.38199 | 0.3 | 0.1 |

Parameters that minimize mean of MAPE

| Mean of MAPE |  |  |
| --- | --- | --- |
| 4.787940 | 0.2 | 0.2 |
| 4.790382 | 0.2 | 0.3 |

Parameters that minimize mean of MAE

| Mean of MAE |  |  |
| --- | --- | --- |
| 2.826612 | 0.2 | 0.2 |
| 2.850716 | 0.2 | 0.3 |

Parameters that minimize var of MSE

| Var of MSE |  |  |
| --- | --- | --- |
| 61.3621 | 0.1 | 0.4 |
| 74.6558 | 0.2 | 0.2 |

Parameters that minimize var of MAPE

| Var of MAPE |  |  |
| --- | --- | --- |
| 3.354422 | 0.1 | 0.4 |
| 4.055937 | 0.2 | 0.2 |

Parameters that minimize var of MAE

| Var of MAE |  |  |
| --- | --- | --- |
| 1.066899 | 0.1 | 0.4 |
| 1.438596 | 0.2 | 0.2 |

The means of all accuracy measures are all minimized by the parameters and , while the variances are all minimized by

#### DMA

resultsDMA <- trainset |> select(close) |>  
 as.data.table() |>   
 fcCV(initialn=12,   
 folds = 20,   
 "DMA",   
 start = 2,  
 end = 6, dist = 1)  
  
aggregateDMA <- resultsDMA[[3]] |>  
 as\_tibble() |> group\_by(M) |>  
 summarise(mean\_MSE=mean(MSE),  
 var\_MSE=var(MSE),  
 mean\_MAPE=mean(MAPE),  
 var\_MAPE=var(MAPE),  
 mean\_MAE=mean(MAE),  
 var\_MAE=var(MAE)) |>  
 ungroup()  
  
kable(head(aggregateDMA),   
 col.names = c("M",  
 "mean (MSE)", "variance (MSE)",  
 "mean (MAPE)", "variance (MAPE)",  
 "mean (MAE)", "variance (MAE)"))

| M | mean (MSE) | variance (MSE) | mean (MAPE) | variance (MAPE) | mean (MAE) | variance (MAE) |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | 25.78217 | 579.8362 | 6.954203 | 16.577070 | 4.084812 | 5.490662 |
| 3 | 30.37991 | 1338.5765 | 7.209714 | 25.216060 | 4.217126 | 9.660735 |
| 4 | 30.57237 | 1369.9274 | 7.312041 | 20.800480 | 4.360110 | 8.813963 |
| 5 | 22.83093 | 538.2283 | 6.547026 | 12.917950 | 3.883035 | 5.390212 |
| 6 | 17.68174 | 398.7954 | 5.672267 | 9.983421 | 3.371007 | 4.064561 |

We find the window size that optimizes each measure:

params <- aggregateDMA |>  
 select(mean\_MSE, mean\_MAPE, mean\_MAE,  
 var\_MSE, var\_MAPE, var\_MAE)   
  
bestparams <- params |>   
 apply(FUN = slice\_min, MARGIN = 2,   
 .data = aggregateDMA, n = 2) |>  
 purrr::map2(names(params), select,   
 M)  
  
titles <- names(bestparams) |>   
 stringr::str\_split(pattern="\_")  
  
for(p in seq(length(bestparams))){  
 kable(bestparams[[p]],   
 caption = paste("Parameters that minimize",  
 titles[[p]][1], "of",  
 titles[[p]][2]),  
 col.names = c(  
 paste(  
 {titles[[p]][1] |>   
 R.utils::capitalize()},   
 "of", titles[[p]][2]),  
 "M"))|> print(p)  
}

Table: Parameters that minimize mean of MSE  
  
| Mean of MSE| M|  
|-----------:|--:|  
| 17.68174| 6|  
| 22.83093| 5|  
  
  
Table: Parameters that minimize mean of MAPE  
  
| Mean of MAPE| M|  
|------------:|--:|  
| 5.672267| 6|  
| 6.547026| 5|  
  
  
Table: Parameters that minimize mean of MAE  
  
| Mean of MAE| M|  
|-----------:|--:|  
| 3.371007| 6|  
| 3.883035| 5|  
  
  
Table: Parameters that minimize var of MSE  
  
| Var of MSE| M|  
|----------:|--:|  
| 398.7954| 6|  
| 538.2283| 5|  
  
  
Table: Parameters that minimize var of MAPE  
  
| Var of MAPE| M|  
|-----------:|--:|  
| 9.983421| 6|  
| 12.917950| 5|  
  
  
Table: Parameters that minimize var of MAE  
  
| Var of MAE| M|  
|----------:|--:|  
| 4.064561| 6|  
| 5.390212| 5|

The optimal parameter seems to be .

### Comparing baselines with cross-validation

We use two methods as a baseline, optimal DES smoothing and auto.arima:

trainset |> select(close) |>   
 stats::HoltWinters(gamma = F)

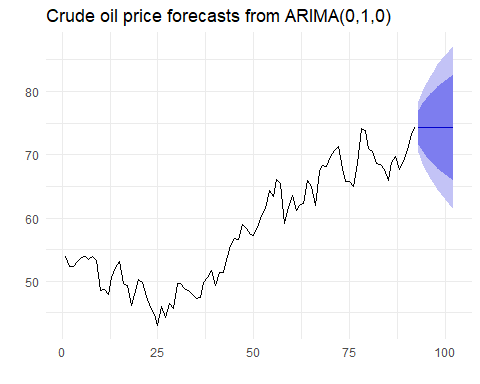
Holt-Winters exponential smoothing with trend and without seasonal component.  
  
Call:  
stats::HoltWinters(x = select(trainset, close), gamma = F)  
  
Smoothing parameters:  
 alpha: 1  
 beta : 0.06980948  
 gamma: FALSE  
  
Coefficients:  
 [,1]  
a 74.3399960  
b 0.4553176

trainset |> select(close) |>   
 forecast::auto.arima()

Series: select(trainset, close)   
ARIMA(0,1,0)   
  
sigma^2 = 4.249: log likelihood = -194.95  
AIC=391.9 AICc=391.94 BIC=394.41

We find that optimal smoothing uses the parameters and , while auto.arima selects a random walk model where the prediction for future observations is based on the last observation available in the training set.

trainset |> select(close) |>   
 auto.arima() |> forecast() |>   
 autoplot() +   
 theme\_minimal() +  
 xlab("") + ylab("") +  
 ggtitle("Crude oil price forecasts from ARIMA(0,1,0)")



Now consider accuracy:

kable(  
 {  
 rbind(  
 trainset |> select(close) |>  
 stats::HoltWinters(gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 forecast::auto.arima() |>   
 predict(4) |> with(pred) |>  
 forecast::accuracy(testset$close)  
 ) |> as\_tibble() |>  
 mutate(method = c("Optimal Smoothing",  
 "Auto ARIMA")) |>  
 tibble::remove\_rownames() |>   
 tibble::column\_to\_rownames(var="method")  
 },  
 col.names=c("ME", "RMSE" ,"MAE", "MPE", "MAPE"),  
 row.names=T  
 )

|  | ME | RMSE | MAE | MPE | MAPE |
| --- | --- | --- | --- | --- | --- |
| Optimal Smoothing | -7.680791 | 8.440171 | 7.680791 | -11.585867 | 11.585867 |
| Auto ARIMA | -6.542497 | 7.198103 | 6.542497 | -9.870582 | 9.870582 |

Compare with the DES smoothing parameters provided by cross-validation, and DMA with and 6.

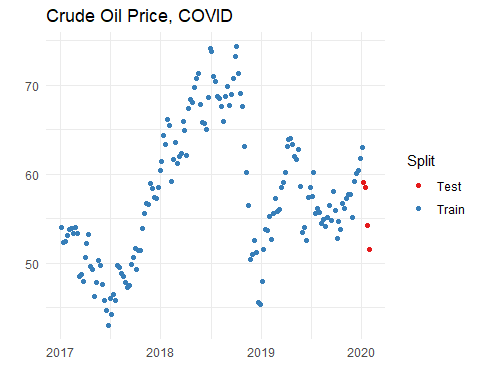
kable(  
 {  
 rbind(  
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 0.2, beta = 0.2,  
 gamma = F) |>   
 predict(4) |> accuracy(testset$close),  
   
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 0.1, beta = 0.4,  
 gamma = F) |>   
 predict(4) |> accuracy(testset$close),  
   
 accuracy({  
 trainset |> select(close) |>   
 smoothCV::dma.dt(m = 6,  
 nahead = 4)}[[2]] |>  
 select(forc) |> ts(), testset$close)  
   
 ) |> as\_tibble() |>  
 mutate(method = c("Minimize means (DES)",  
 "Minimize variance (DES)",  
 "Optimal DMA"  
 )) |>  
 tibble::remove\_rownames() |>   
 tibble::column\_to\_rownames(var="method")  
 },  
 col.names=c("ME", "RMSE" ,"MAE", "MPE", "MAPE"),  
 row.names=T  
 )

|  | ME | RMSE | MAE | MPE | MAPE |
| --- | --- | --- | --- | --- | --- |
| Minimize means (DES) | -3.580955 | 4.863738 | 3.761038 | -5.512808 | 5.765238 |
| Minimize variance (DES) | -2.113257 | 3.718060 | 2.867624 | -3.328233 | 4.385658 |
| Optimal DMA | -6.741388 | 7.746467 | 6.741388 | -10.218375 | 10.218375 |

DES parameters that minimize variance of error measures performed best, followed by DES parameters that minimize mean of error measures. Both performed better than baseline methods and DMA. DMA at performed worse than auto.arima. Optimal DES performed worst.

## Case the Second: COVID

trainset <- crude\_wk |>   
 filter(date <= as.Date("2020-01-07")) |>  
 mutate(Split = "Train")  
  
testset <- crude\_wk |>   
 filter(date > as.Date("2020-01-07")) |>   
 mutate(Split = "Test") |> head(4)  
  
bind\_rows(trainset,testset) |>   
 ggplot(aes(x = date, y = close,   
 color = Split)) +  
 geom\_point() +  
 scale\_color\_brewer(palette="Set1") +  
 theme\_minimal() +  
 xlab ("") + ylab ("") +  
 ggtitle("Crude Oil Price, COVID")



Scientists in China announced a new coronavirus in January 7, 2020. We see this leads to a sharp downward trend.

### Aggregation

#### DES

Now we optimize parameters in the training set through cross-validation, and generate an aggregation table for DES:

resultsDES <- trainset |> select(close) |>  
 as.data.table() |>   
 fcCV(fullset = \_,   
 initialn= 37,   
 folds = 30,   
 "DES",   
 alphrange=seq(0.1,1,0.1),   
 betarange=seq(0.1,1,0.1))  
  
aggregateDES <- resultsDES[[3]] |>  
 as\_tibble() |>  
 group\_by(alphrange, betarange) |>  
 summarise(mean\_MSE=mean(MSE),  
 var\_MSE=var(MSE),  
 mean\_MAPE=mean(MAPE),  
 var\_MAPE=var(MAPE),  
 mean\_MAE=mean(MAE),  
 var\_MAE=var(MAE)) |>  
 ungroup()  
  
kable(head(aggregateDES),   
 col.names = c("$\\alpha$", "$\\beta$",  
 "mean (MSE)", "variance (MSE)",  
 "mean (MAPE)", "variance (MAPE)",  
 "mean (MAE)", "variance (MAE)"))

|  |  | mean (MSE) | variance (MSE) | mean (MAPE) | variance (MAPE) | mean (MAE) | variance (MAE) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0.1 | 0.1 | 42.49662 | 4417.965 | 8.624361 | 55.69392 | 4.991226 | 15.75048 |
| 0.1 | 0.2 | 49.45354 | 4100.300 | 9.623423 | 56.31495 | 5.550512 | 16.29361 |
| 0.1 | 0.3 | 56.69207 | 6037.271 | 10.336899 | 65.88790 | 5.945288 | 18.56582 |
| 0.1 | 0.4 | 56.57830 | 7947.054 | 9.700452 | 78.86570 | 5.583893 | 22.64872 |
| 0.1 | 0.5 | 56.12909 | 7976.801 | 9.350485 | 82.51669 | 5.399909 | 24.07116 |
| 0.1 | 0.6 | 57.35164 | 7962.679 | 9.651215 | 78.44395 | 5.598998 | 22.93820 |

We find parameters that optimize each measure:

params <- aggregateDES |>  
 select(mean\_MSE, mean\_MAPE, mean\_MAE,  
 var\_MSE, var\_MAPE, var\_MAE)   
  
bestparams <- params |>   
 apply(FUN = slice\_min, MARGIN = 2,   
 .data = aggregateDES, n = 2) |>  
 purrr::map2(names(params), select,   
 alphrange, betarange)  
  
titles <- names(bestparams) |>   
 stringr::str\_split(pattern="\_")  
  
for(p in seq(length(bestparams))){  
 paste("Parameters that minimize",   
 titles[[p]][1], "of", titles[[p]][2]) |>  
 kable(bestparams[[p]],   
 caption = \_,  
 col.names = c(  
 paste(  
 {titles[[p]][1] |>  
 R.utils::capitalize()},   
 "of", titles[[p]][2]  
 )  
 ,"$\\alpha$", "$\\beta$"))|> print(p)  
}

Parameters that minimize mean of MSE

| Mean of MSE |  |  |
| --- | --- | --- |
| 10.45230 | 1 | 0.4 |
| 10.63281 | 1 | 0.3 |

Parameters that minimize mean of MAPE

| Mean of MAPE |  |  |
| --- | --- | --- |
| 4.181675 | 1 | 0.3 |
| 4.239957 | 1 | 0.2 |

Parameters that minimize mean of MAE

| Mean of MAE |  |  |
| --- | --- | --- |
| 2.501439 | 1.0 | 0.3 |
| 2.537278 | 0.9 | 0.4 |

Parameters that minimize var of MSE

| Var of MSE |  |  |
| --- | --- | --- |
| 95.69682 | 1.0 | 0.5 |
| 105.82153 | 0.9 | 0.6 |

Parameters that minimize var of MAPE

| Var of MAPE |  |  |
| --- | --- | --- |
| 5.278833 | 1 | 0.5 |
| 5.597181 | 1 | 0.4 |

Parameters that minimize var of MAE

| Var of MAE |  |  |
| --- | --- | --- |
| 1.741283 | 1.0 | 0.5 |
| 1.968133 | 0.9 | 0.6 |

Mean of MSE is minimized by the parameters , . Means of other accuracy measures are minimized by the set of parameters and , while the variance of all measures are minimized by the parameters and .

#### DMA

Meanwhile for DMA:

resultsDMA <- trainset |> select(close) |>  
 as.data.table() |>   
 fcCV(initialn=37,   
 folds = 30,   
 "DMA",   
 start = 2,  
 end = 18, dist = 1)  
  
aggregateDMA <- resultsDMA[[3]] |>  
 as\_tibble() |> group\_by(M) |>  
 summarise(mean\_MSE=mean(MSE),  
 var\_MSE=var(MSE),  
 mean\_MAPE=mean(MAPE),  
 var\_MAPE=var(MAPE),  
 mean\_MAE=mean(MAE),  
 var\_MAE=var(MAE)) |>  
 ungroup()  
  
kable(head(aggregateDMA),   
 col.names = c("M",  
 "mean (MSE)", "variance (MSE)",  
 "mean (MAPE)", "variance (MAPE)",  
 "mean (MAE)", "variance (MAE)"))

| M | mean (MSE) | variance (MSE) | mean (MAPE) | variance (MAPE) | mean (MAE) | variance (MAE) |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | 14.52212 | 260.9809 | 4.983072 | 10.95377 | 2.946375 | 3.297824 |
| 3 | 21.51417 | 1319.2832 | 5.806369 | 20.61085 | 3.455916 | 6.771077 |
| 4 | 31.58821 | 1510.9497 | 7.373402 | 25.99712 | 4.413767 | 9.428453 |
| 5 | 32.64612 | 1347.2257 | 7.672119 | 24.96457 | 4.598170 | 8.964734 |
| 6 | 30.61168 | 1492.6508 | 7.248117 | 30.92910 | 4.307940 | 9.760991 |
| 7 | 34.06831 | 3099.7276 | 7.310130 | 44.76550 | 4.306901 | 13.366880 |

We find the window size that optimizes each measure:

params <- aggregateDMA |>  
 select(mean\_MSE, mean\_MAPE, mean\_MAE,  
 var\_MSE, var\_MAPE, var\_MAE)   
  
bestparams <- params |>   
 apply(FUN = slice\_min, MARGIN = 2,   
 .data = aggregateDMA, n = 2) |>  
 purrr::map2(names(params), select,   
 M)  
  
titles <- names(bestparams) |>   
 stringr::str\_split(pattern="\_")  
  
for(p in seq(length(bestparams))){  
 kable(bestparams[[p]],   
 caption = paste("Parameters that minimize",  
 titles[[p]][1], "of",  
 titles[[p]][2]),  
 col.names = c(  
 paste(  
 {titles[[p]][1] |>   
 R.utils::capitalize()},   
 "of", titles[[p]][2]),  
 "M"))|> print(p)  
}

Parameters that minimize mean of MSE

| Mean of MSE | M |
| --- | --- |
| 14.52212 | 2 |
| 21.51417 | 3 |

Parameters that minimize mean of MAPE

| Mean of MAPE | M |
| --- | --- |
| 4.983072 | 2 |
| 5.806369 | 3 |

Parameters that minimize mean of MAE

| Mean of MAE | M |
| --- | --- |
| 2.946375 | 2 |
| 3.455916 | 3 |

Parameters that minimize var of MSE

| Var of MSE | M |
| --- | --- |
| 260.9809 | 2 |
| 1319.2832 | 3 |

Parameters that minimize var of MAPE

| Var of MAPE | M |
| --- | --- |
| 10.95377 | 2 |
| 20.61085 | 3 |

Parameters that minimize var of MAE

| Var of MAE | M |
| --- | --- |
| 3.297824 | 2 |
| 6.771077 | 3 |

The optimal parameter seems to be .

### Comparing baselines with cross-validation

We use two methods as a baseline:

trainset |> select(close) |>  
 auto.arima()

Series: select(trainset, close)   
ARIMA(0,1,0)   
  
sigma^2 = 4.878: log likelihood = -344.96  
AIC=691.92 AICc=691.95 BIC=694.97

trainset |> select(close) |>  
 stats::HoltWinters(gamma=F)

Holt-Winters exponential smoothing with trend and without seasonal component.  
  
Call:  
stats::HoltWinters(x = select(trainset, close), gamma = F)  
  
Smoothing parameters:  
 alpha: 1  
 beta : 0.06321651  
 gamma: FALSE  
  
Coefficients:  
 [,1]  
a 63.049999  
b 0.352854

As before, auto.arima picked a random walk model. Moreover optimal smoothing picked the parameters and

kable(  
 {  
 rbind(  
 trainset |> select(close) |>  
 stats::HoltWinters(gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 forecast::auto.arima() |>   
 predict(4) |> with(pred) |>  
 forecast::accuracy(testset$close)  
 ) |> as\_tibble() |>  
 mutate(method = c("Optimal Smoothing",  
 "Auto ARIMA")) |>  
 tibble::remove\_rownames() |>   
 tibble::column\_to\_rownames(var="method")  
 },  
 col.names=c("ME", "RMSE" ,"MAE", "MPE", "MAPE"),  
 row.names=T  
 )

|  | ME | RMSE | MAE | MPE | MAPE |
| --- | --- | --- | --- | --- | --- |
| Optimal Smoothing | -8.099633 | 8.818481 | 8.099633 | -14.90619 | 14.90619 |
| Auto ARIMA | -7.217498 | 7.857159 | 7.217498 | -13.28268 | 13.28268 |

Compare with the DES smoothing parameters provided by cross-validation and DMA.

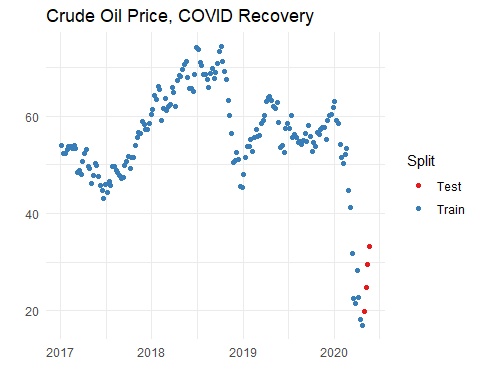
kable(  
 {  
 rbind(  
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 1, beta = 0.4,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 1, beta = 0.3,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 1, beta = 0.5,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 accuracy({  
 trainset |> select(close) |>   
 smoothCV::dma.dt(m = 2,  
 nahead = 4)}[[2]] |>  
 select(forc) |> ts(), testset$close)  
   
   
 ) |> as\_tibble() |>  
 mutate(method = c("Minimize mean MSE",  
 "Minimize mean MAPE and MAE",  
 "Minimize variances",  
 "Optimal DMA"  
 )) |>  
 tibble::remove\_rownames() |>   
 tibble::column\_to\_rownames(var="method")  
 },  
 col.names=c("ME", "RMSE" ,"MAE", "MPE", "MAPE"),  
 row.names=T  
 )

|  | ME | RMSE | MAE | MPE | MAPE |
| --- | --- | --- | --- | --- | --- |
| Minimize mean MSE | -9.999082 | 10.89145 | 9.999082 | -18.40198 | 18.40198 |
| Minimize mean MAPE and MAE | -9.752956 | 10.62267 | 9.752956 | -17.94900 | 17.94900 |
| Minimize variances | -10.157459 | 11.06442 | 10.157459 | -18.69346 | 18.69346 |
| Optimal DMA | -10.467500 | 11.40527 | 10.467500 | -19.26461 | 19.26461 |

Cross-validation performed worse than the baseline.

## Case the Third: COVID Recovery

trainset <- crude\_wk |>   
 filter(date <= as.Date("2020-04-27")) |>  
 mutate(Split = "Train")  
testset <- crude\_wk |>   
 filter(date > as.Date("2020-04-27")) |>  
 head(4) |> mutate(Split = "Test")  
  
bind\_rows(trainset,testset) |>  
ggplot(aes(x = date, y = close,   
 color = Split)) +  
 geom\_point() +  
 scale\_color\_brewer(palette="Set1") +  
 theme\_minimal() +  
 xlab ("") + ylab ("") +  
 ggtitle("Crude Oil Price, COVID Recovery")



Saudi Arabia and Russia agreed to cut production in April 2020. However, the effect of these cuts on WTI futures price were lagged - supply needed time to adjust. Therefore, oil prices only began to rise in May. This is another area where we can experiment with smoothing.

### Aggregation

#### DES

Now we optimize parameters in the training set through cross-validation, and generate an aggregation table:

resultsDES <- trainset |> select(close) |>  
 as.data.table() |>   
 fcCV(fullset = \_,   
 initialn=53,   
 folds = 30,   
 "DES",   
 alphrange=seq(0.1,1,0.1),   
 betarange=seq(0.1,1,0.1))  
  
aggregateDES <- resultsDES[[3]] |>  
 as\_tibble() |>  
 group\_by(alphrange, betarange) |>  
 summarise(mean\_MSE=mean(MSE),  
 var\_MSE=var(MSE),  
 mean\_MAPE=mean(MAPE),  
 var\_MAPE=var(MAPE),  
 mean\_MAE=mean(MAE),  
 var\_MAE=var(MAE)) |>  
 ungroup()  
  
kable(head(aggregateDES),   
 col.names = c("$\\alpha$", "$\\beta$",  
 "mean (MSE)", "variance (MSE)",  
 "mean (MAPE)", "variance (MAPE)",  
 "mean (MAE)", "variance (MAE)"))

|  |  | mean (MSE) | variance (MSE) | mean (MAPE) | variance (MAPE) | mean (MAE) | variance (MAE) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0.1 | 0.1 | 79.58787 | 22409.63 | 14.95391 | 584.7713 | 6.288107 | 36.01184 |
| 0.1 | 0.2 | 84.09288 | 19081.28 | 15.81616 | 474.2507 | 6.980449 | 30.88001 |
| 0.1 | 0.3 | 91.22610 | 20100.20 | 16.18864 | 393.1239 | 7.457094 | 30.60582 |
| 0.1 | 0.4 | 81.53257 | 14612.62 | 14.03686 | 268.6871 | 6.822230 | 30.31566 |
| 0.1 | 0.5 | 73.28523 | 11113.82 | 12.71666 | 211.2417 | 6.372895 | 28.06219 |
| 0.1 | 0.6 | 73.02217 | 10746.59 | 12.86009 | 197.2549 | 6.504965 | 25.75470 |

We find parameters that optimize each measure:

params <- aggregateDES |>  
 select(mean\_MSE, mean\_MAPE, mean\_MAE,  
 var\_MSE, var\_MAPE, var\_MAE)   
  
bestparams <- params |>   
 apply(FUN = slice\_min, MARGIN = 2,   
 .data = aggregateDES, n = 2) |>  
 purrr::map2(names(params), select,   
 alphrange, betarange)  
  
titles <- names(bestparams) |>   
 stringr::str\_split(pattern="\_")  
  
for(p in seq(length(bestparams))){  
 paste("Parameters that minimize",   
 titles[[p]][1], "of", titles[[p]][2]) |>  
 kable(bestparams[[p]], caption = \_,  
 col.names = c(  
 paste(  
 {titles[[p]][1] |> R.utils::capitalize()},   
 "of", titles[[p]][2]  
 )  
 ,"$\\alpha$", "$\\beta$"))|> print(p)  
}

Parameters that minimize mean of MSE

| Mean of MSE |  |  |
| --- | --- | --- |
| 21.38116 | 1 | 0.5 |
| 21.77953 | 1 | 0.6 |

Parameters that minimize mean of MAPE

| Mean of MAPE |  |  |
| --- | --- | --- |
| 7.250763 | 1.0 | 0.7 |
| 7.366123 | 0.9 | 0.9 |

Parameters that minimize mean of MAE

| Mean of MAE |  |  |
| --- | --- | --- |
| 3.407517 | 1 | 0.3 |
| 3.416105 | 1 | 0.4 |

Parameters that minimize var of MSE

| Var of MSE |  |  |
| --- | --- | --- |
| 796.0093 | 1 | 0.6 |
| 815.0225 | 1 | 0.7 |

Parameters that minimize var of MAPE

| Var of MAPE |  |  |
| --- | --- | --- |
| 34.91936 | 1.0 | 0.9 |
| 40.65303 | 0.9 | 1.0 |

Parameters that minimize var of MAE

| Var of MAE |  |  |
| --- | --- | --- |
| 5.631918 | 1 | 0.6 |
| 5.977887 | 1 | 0.5 |

Mean of MSE is minimized by the parameters and , mean of MAPE is minimized by the parameters and , and mean of MAE is minimized by the parameters and . The parameters and minimize variance of MSE and MAE, while the parameters and .

#### DMA

Meanwhile for DMA:

resultsDMA <- trainset |> select(close) |>  
 as.data.table() |>   
 fcCV(initialn=53,   
 folds = 30,   
 "DMA",   
 start = 2,  
 end = 26, dist = 1)  
  
aggregateDMA <- resultsDMA[[3]] |>  
 as\_tibble() |> group\_by(M) |>  
 summarise(mean\_MSE=mean(MSE),  
 var\_MSE=var(MSE),  
 mean\_MAPE=mean(MAPE),  
 var\_MAPE=var(MAPE),  
 mean\_MAE=mean(MAE),  
 var\_MAE=var(MAE)) |>  
 ungroup()  
  
kable(head(aggregateDMA),   
 col.names = c("M",  
 "mean (MSE)", "variance (MSE)",  
 "mean (MAPE)", "variance (MAPE)",  
 "mean (MAE)", "variance (MAE)"))

| M | mean (MSE) | variance (MSE) | mean (MAPE) | variance (MAPE) | mean (MAE) | variance (MAE) |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | 30.00932 | 2363.821 | 9.030868 | 198.6425 | 3.920375 | 11.08079 |
| 3 | 50.60450 | 9959.806 | 11.816413 | 442.1122 | 4.861814 | 22.03833 |
| 4 | 56.37461 | 6538.577 | 12.823202 | 335.2772 | 5.726048 | 18.65502 |
| 5 | 49.71152 | 3263.001 | 12.035050 | 199.3164 | 5.664199 | 13.16143 |
| 6 | 41.54493 | 2459.919 | 10.287268 | 111.6493 | 5.073461 | 11.58342 |
| 7 | 45.80808 | 4530.937 | 9.992543 | 129.5905 | 5.043030 | 16.22658 |

We find the window size that optimizes each measure:

params <- aggregateDMA |>  
 select(mean\_MSE, mean\_MAPE, mean\_MAE,  
 var\_MSE, var\_MAPE, var\_MAE)   
  
bestparams <- params |>   
 apply(FUN = slice\_min, MARGIN = 2,   
 .data = aggregateDMA, n = 2) |>  
 purrr::map2(names(params), select,   
 M)  
  
titles <- names(bestparams) |>   
 stringr::str\_split(pattern="\_")  
  
for(p in seq(length(bestparams))){  
 kable(bestparams[[p]],   
 caption = paste("Parameters that minimize",  
 titles[[p]][1], "of",  
 titles[[p]][2]),  
 col.names = c(  
 paste(  
 {titles[[p]][1] |>   
 R.utils::capitalize()},   
 "of", titles[[p]][2]),  
 "M"))|> print(p)  
}

Parameters that minimize mean of MSE

| Mean of MSE | M |
| --- | --- |
| 30.00932 | 2 |
| 41.54493 | 6 |

Parameters that minimize mean of MAPE

| Mean of MAPE | M |
| --- | --- |
| 9.030868 | 2 |
| 9.992543 | 7 |

Parameters that minimize mean of MAE

| Mean of MAE | M |
| --- | --- |
| 3.920375 | 2 |
| 4.861814 | 3 |

Parameters that minimize var of MSE

| Var of MSE | M |
| --- | --- |
| 2363.821 | 2 |
| 2459.919 | 6 |

Parameters that minimize var of MAPE

| Var of MAPE | M |
| --- | --- |
| 111.6493 | 6 |
| 129.5905 | 7 |

Parameters that minimize var of MAE

| Var of MAE | M |
| --- | --- |
| 11.08079 | 2 |
| 11.58342 | 6 |

The optimal parameter seems to be or 6.

### Comparison

We use two methods as a baseline:

trainset |> select(close) |>  
 auto.arima()

Series: select(trainset, close)   
ARIMA(0,1,1)   
  
Coefficients:  
 ma1  
 0.1731  
s.e. 0.0760  
  
sigma^2 = 6.666: log likelihood = -406.71  
AIC=817.42 AICc=817.49 BIC=823.71

trainset |> select(close) |>  
 stats::HoltWinters(gamma=F)

Holt-Winters exponential smoothing with trend and without seasonal component.  
  
Call:  
stats::HoltWinters(x = select(trainset, close), gamma = F)  
  
Smoothing parameters:  
 alpha: 1  
 beta : 0.08341341  
 gamma: FALSE  
  
Coefficients:  
 [,1]  
a 16.940001  
b -2.062455

auto.arima picked a MA(1) model. Moreover optimal smoothing picked the parameters and

kable(  
 {  
 rbind(  
 trainset |> select(close) |>  
 stats::HoltWinters(gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 forecast::auto.arima() |>   
 predict(4) |> with(pred) |>  
 forecast::accuracy(testset$close),  
   
 {trainset |> select(close) |>  
 naive(4) |>  
 forecast::accuracy(testset$close)  
 }[2,-(6:7)]  
   
 ) |> as\_tibble() |>  
 mutate(method = c("Optimal Smoothing",  
 "Auto ARIMA",  
 "Naive")) |>  
 tibble::remove\_rownames() |>   
 tibble::column\_to\_rownames(var="method")  
 },  
 col.names=c("ME", "RMSE" ,"MAE", "MPE", "MAPE"),  
 row.names=T  
 )

|  | ME | RMSE | MAE | MPE | MAPE |
| --- | --- | --- | --- | --- | --- |
| Optimal Smoothing | 15.016138 | 16.72024 | 15.016138 | 52.57841 | 52.57841 |
| Auto ARIMA | 9.990698 | 11.19486 | 9.990698 | 34.85109 | 34.85109 |
| Naive | 9.859999 | 11.07838 | 9.859999 | 34.34453 | 34.34453 |

Compare with the DES smoothing parameters provided by cross-validation.

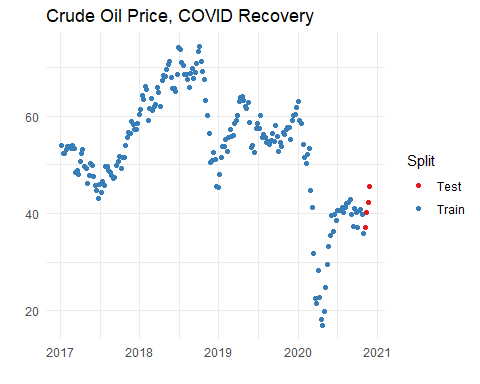
kable(  
 {  
 rbind(  
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 1, beta = 0.5,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 1, beta = 0.7,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 1, beta = 0.3,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 1, beta = 0.6,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 1, beta = 0.9,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 accuracy({  
 trainset |> select(close) |>   
 smoothCV::dma.dt(m = 2,  
 nahead = 4)}[[2]] |>  
 select(forc) |> ts(), testset$close)  
   
 ) |> as\_tibble() |>  
 mutate(method = c("Minimize mean MSE",  
 "Minimize mean MAPE",  
 "Minimize mean MAE",  
 "Minimize var MSE and MAE",  
 "Minimize var MAPE",  
 "DMA"  
 )) |>  
 tibble::remove\_rownames() |>   
 tibble::column\_to\_rownames(var="method")  
 },  
 col.names=c("ME", "RMSE" ,"MAE", "MPE", "MAPE"),  
 row.names=T  
 )

|  | ME | RMSE | MAE | MPE | MAPE |
| --- | --- | --- | --- | --- | --- |
| Minimize mean MSE | 15.70008 | 17.46892 | 15.70008 | 54.99705 | 54.99705 |
| Minimize mean MAPE | 15.16792 | 16.88639 | 15.16792 | 53.11516 | 53.11516 |
| Minimize mean MAE | 16.28543 | 18.10972 | 16.28543 | 57.06707 | 57.06707 |
| Minimize var MSE and MAE | 15.48580 | 17.23435 | 15.48580 | 54.23928 | 54.23928 |
| Minimize var MAPE | 13.97337 | 15.57887 | 13.97337 | 48.89082 | 48.89082 |
| DMA | 17.92500 | 19.75378 | 17.92500 | 63.13329 | 63.13329 |

auto.arima performed best. Cross validation by minimizing variance of MAPE performed best among all smoothing methods, DMA performed worst. Optimal smoothing performed better than all cross-validation parameters except the one that minimizies variance of MAPE. However, the performance between these models are quite close.

## Case the Fourth: Vaccines

trainset <- crude\_wk |>   
 filter(date <= as.Date("2020-11-01")) |>  
 mutate(Split = "Train")  
testset <- crude\_wk |>   
 filter(date > as.Date("2020-11-01")) |>  
 head(4) |> mutate(Split = "Test")  
bind\_rows(trainset,testset) |>  
 ggplot(aes(x = date, y = close,   
 color = Split)) +  
 geom\_point() +  
 scale\_color\_brewer(palette="Set1") +  
 theme\_minimal() +  
 xlab ("") + ylab ("") +  
 ggtitle("Crude Oil Price, COVID Recovery")



Pfizer announced a COVID-19 vaccine in November 8, 2020, leading to a sharp rise in oil futures price.

### Aggregation

#### DES

Now we optimize parameters in the training set through cross-validation, and generate an aggregation table:

resultsDES <- trainset |> select(close) |>  
 as.data.table() |>   
 fcCV(fullset = \_,   
 initialn=80,   
 folds = 30,   
 "DES",   
 alphrange=seq(0.1,1,0.1),   
 betarange=seq(0.1,1,0.1))  
  
aggregateDES <- resultsDES[[3]] |>  
 as\_tibble() |>  
 group\_by(alphrange, betarange) |>  
 summarise(mean\_MSE=mean(MSE),  
 var\_MSE=var(MSE),  
 mean\_MAPE=mean(MAPE),  
 var\_MAPE=var(MAPE),  
 mean\_MAE=mean(MAE),  
 var\_MAE=var(MAE)) |>  
 ungroup()  
  
kable(head(aggregateDES),   
 col.names = c("$\\alpha$", "$\\beta$",  
 "mean (MSE)", "variance (MSE)",  
 "mean (MAPE)", "variance (MAPE)",  
 "mean (MAE)", "variance (MAE)"))

|  |  | mean (MSE) | variance (MSE) | mean (MAPE) | variance (MAPE) | mean (MAE) | variance (MAE) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0.1 | 0.1 | 112.0335 | 22611.52 | 20.21526 | 557.5587 | 7.992590 | 42.13256 |
| 0.1 | 0.2 | 137.3531 | 33536.82 | 22.49059 | 536.0381 | 9.215490 | 45.98062 |
| 0.1 | 0.3 | 173.6070 | 59419.65 | 24.85633 | 584.0476 | 10.352482 | 58.05164 |
| 0.1 | 0.4 | 176.4699 | 70676.68 | 24.09048 | 547.5882 | 10.143880 | 63.05238 |
| 0.1 | 0.5 | 162.5202 | 62639.72 | 22.40748 | 515.8486 | 9.474089 | 61.59452 |
| 0.1 | 0.6 | 160.4935 | 58306.55 | 22.60598 | 496.7887 | 9.611769 | 55.87709 |

We find parameters that optimize each measure:

params <- aggregateDES |>  
 select(mean\_MSE, mean\_MAPE, mean\_MAE,  
 var\_MSE, var\_MAPE, var\_MAE)   
  
bestparams <- params |>   
 apply(FUN = slice\_min, MARGIN = 2,   
 .data = aggregateDES, n = 2) |>  
 purrr::map2(names(params), select,   
 alphrange, betarange)  
  
titles <- names(bestparams) |>   
 stringr::str\_split(pattern="\_")  
  
for(p in seq(length(bestparams))){  
 paste("Parameters that minimize",   
 titles[[p]][1], "of", titles[[p]][2]) |>  
 kable(bestparams[[p]], caption = \_,  
 col.names = c(  
 paste(  
 {titles[[p]][1] |> R.utils::capitalize()},   
 "of", titles[[p]][2]  
 )  
 ,"$\\alpha$", "$\\beta$"))|> print(p)  
}

Parameters that minimize mean of MSE

| Mean of MSE |  |  |
| --- | --- | --- |
| 42.16412 | 0.8 | 0.1 |
| 42.18678 | 0.7 | 0.1 |

Parameters that minimize mean of MAPE

| Mean of MAPE |  |  |
| --- | --- | --- |
| 10.59267 | 0.7 | 0.1 |
| 10.61026 | 0.6 | 0.1 |

Parameters that minimize mean of MAE

| Mean of MAE |  |  |
| --- | --- | --- |
| 4.576704 | 0.8 | 0.1 |
| 4.579595 | 0.7 | 0.1 |

Parameters that minimize var of MSE

| Var of MSE |  |  |
| --- | --- | --- |
| 3925.311 | 0.4 | 0.5 |
| 4012.273 | 0.4 | 0.4 |

Parameters that minimize var of MAPE

| Var of MAPE |  |  |
| --- | --- | --- |
| 151.4943 | 0.7 | 0.1 |
| 152.7987 | 0.6 | 0.1 |

Parameters that minimize var of MAE

| Var of MAE |  |  |
| --- | --- | --- |
| 13.75327 | 0.5 | 0.3 |
| 13.79796 | 0.5 | 0.4 |

Means of MSE and MAE are minimized by the parameters and , mean and variance of MAPE are minimized by the parameters and . Variance of MSE is minimized by and and variance of MAE is minimized by and .

#### DMA

Meanwhile for DMA:

resultsDMA <- trainset |> select(close) |>  
 as.data.table() |>   
 fcCV(initialn=80,   
 folds = 30,   
 "DMA",   
 start = 2,  
 end = 39, dist = 1)  
  
aggregateDMA <- resultsDMA[[3]] |>  
 as\_tibble() |> group\_by(M) |>  
 summarise(mean\_MSE=mean(MSE),  
 var\_MSE=var(MSE),  
 mean\_MAPE=mean(MAPE),  
 var\_MAPE=var(MAPE),  
 mean\_MAE=mean(MAE),  
 var\_MAE=var(MAE)) |>  
 ungroup()  
  
kable(head(aggregateDMA),   
 col.names = c("M",  
 "mean (MSE)", "variance (MSE)",  
 "mean (MAPE)", "variance (MAPE)",  
 "mean (MAE)", "variance (MAE)"))

| M | mean (MSE) | variance (MSE) | mean (MAPE) | variance (MAPE) | mean (MAE) | variance (MAE) |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | 92.18967 | 28846.416 | 16.67993 | 609.9523 | 6.447333 | 36.23488 |
| 3 | 59.79302 | 9141.685 | 12.72939 | 273.0482 | 5.407204 | 21.84371 |
| 4 | 51.42964 | 3319.927 | 12.67614 | 145.9223 | 5.522441 | 12.77438 |
| 5 | 52.79086 | 5113.880 | 12.82427 | 183.4710 | 5.476253 | 15.49009 |
| 6 | 64.42980 | 10191.813 | 13.69626 | 262.7885 | 5.789843 | 22.84463 |
| 7 | 70.11533 | 11712.865 | 14.12517 | 268.7640 | 5.983437 | 26.25276 |

We find the window size that optimizes each measure:

params <- aggregateDMA |>  
 select(mean\_MSE, mean\_MAPE, mean\_MAE,  
 var\_MSE, var\_MAPE, var\_MAE)   
  
bestparams <- params |>   
 apply(FUN = slice\_min, MARGIN = 2,   
 .data = aggregateDMA, n = 2) |>  
 purrr::map2(names(params), select,   
 M)  
  
titles <- names(bestparams) |>   
 stringr::str\_split(pattern="\_")  
  
for(p in seq(length(bestparams))){  
 kable(bestparams[[p]],   
 caption = paste("Parameters that minimize",  
 titles[[p]][1], "of",  
 titles[[p]][2]),  
 col.names = c(  
 paste(  
 {titles[[p]][1] |>   
 R.utils::capitalize()},   
 "of", titles[[p]][2]),  
 "M"))|> print(p)  
}

Parameters that minimize mean of MSE

| Mean of MSE | M |
| --- | --- |
| 51.42964 | 4 |
| 52.79086 | 5 |

Parameters that minimize mean of MAPE

| Mean of MAPE | M |
| --- | --- |
| 12.67614 | 4 |
| 12.72939 | 3 |

Parameters that minimize mean of MAE

| Mean of MAE | M |
| --- | --- |
| 5.407204 | 3 |
| 5.476253 | 5 |

Parameters that minimize var of MSE

| Var of MSE | M |
| --- | --- |
| 3319.927 | 4 |
| 5113.880 | 5 |

Parameters that minimize var of MAPE

| Var of MAPE | M |
| --- | --- |
| 145.9223 | 4 |
| 183.4710 | 5 |

Parameters that minimize var of MAE

| Var of MAE | M |
| --- | --- |
| 12.77438 | 4 |
| 15.49009 | 5 |

The optimal parameter seems to be or 3.

### Comparison

We use three methods as a baseline:

trainset |> select(close) |>  
 auto.arima()

Series: select(trainset, close)   
ARIMA(0,1,1)   
  
Coefficients:  
 ma1  
 0.1627  
s.e. 0.0694  
  
sigma^2 = 6.731: log likelihood = -471.61  
AIC=947.21 AICc=947.27 BIC=953.8

trainset |> select(close) |>  
 stats::HoltWinters(gamma=F)

Holt-Winters exponential smoothing with trend and without seasonal component.  
  
Call:  
stats::HoltWinters(x = select(trainset, close), gamma = F)  
  
Smoothing parameters:  
 alpha: 1  
 beta : 0.0557182  
 gamma: FALSE  
  
Coefficients:  
 [,1]  
a 35.7900010  
b -0.2503346

auto.arima chooses a MA(1) model. We also choose the naive method and optimal DES

kable(  
 {  
 rbind(  
 trainset |> select(close) |>  
 stats::HoltWinters(gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 forecast::auto.arima() |>   
 predict(4) |> with(pred) |>  
 forecast::accuracy(testset$close),  
   
 {trainset |> select(close) |>  
 naive(4) |>  
 forecast::accuracy(testset$close)  
 }[2,-(6:7)]  
   
 ) |> as\_tibble() |>  
 mutate(method = c("Optimal Smoothing",  
 "Auto ARIMA",  
 "Naive")) |>  
 tibble::remove\_rownames() |>   
 tibble::column\_to\_rownames(var="method")  
 },  
 col.names=c("ME", "RMSE" ,"MAE", "MPE", "MAPE"),  
 row.names=T  
 )

|  | ME | RMSE | MAE | MPE | MAPE |
| --- | --- | --- | --- | --- | --- |
| Optimal Smoothing | 6.073336 | 6.927046 | 6.073336 | 14.20847 | 14.20847 |
| Auto ARIMA | 6.082360 | 6.805415 | 6.082360 | 14.28079 | 14.28079 |
| Naive | 5.447499 | 6.244503 | 5.447499 | 12.73280 | 12.73280 |

Compare with the DES smoothing parameters provided by cross-validation. Variance of MSE is minmized

kable(  
 {  
 rbind(  
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 0.8, beta = 0.1,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 0.7, beta = 0.1,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 0.4, beta = 0.5,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 trainset |> select(close) |>  
 stats::HoltWinters(alpha = 0.5, beta = 0.3,  
 gamma = F) |>   
 predict(4) |>  
 forecast::accuracy(testset$close),  
   
 accuracy({  
 trainset |> select(close) |>   
 smoothCV::dma.dt(m = 4,  
 nahead = 4)}[[2]] |>  
 select(forc) |> ts(), testset$close),  
   
 accuracy({  
 trainset |> select(close) |>   
 smoothCV::dma.dt(m = 3,  
 nahead = 4)}[[2]] |>  
 select(forc) |> ts(), testset$close)  
   
 ) |> as\_tibble() |>  
 mutate(method = c("Minimize mean MSE and MAE",  
 "Minimize mean and var MAPE",  
 "Minimize var MSE",  
 "Minimize var MAE",  
 "DMA (1)",  
 "DMA(2)"  
 )) |>  
 tibble::remove\_rownames() |>   
 tibble::column\_to\_rownames(var="method")  
 },  
 col.names=c("ME", "RMSE" ,"MAE", "MPE", "MAPE"),  
 row.names=T  
 )

|  | ME | RMSE | MAE | MPE | MAPE |
| --- | --- | --- | --- | --- | --- |
| Minimize mean MSE and MAE | 5.289692 | 6.270778 | 5.289692 | 12.291150 | 12.291150 |
| Minimize mean and var MAPE | 4.725506 | 5.773491 | 4.725506 | 10.924638 | 10.924638 |
| Minimize var MSE | 4.340666 | 5.709481 | 4.660910 | 9.915592 | 10.777853 |
| Minimize var MAE | 4.747951 | 6.006911 | 4.844808 | 10.913969 | 11.174760 |
| DMA (1) | 2.759167 | 4.283003 | 3.578647 | 6.137524 | 8.343985 |
| DMA(2) | 5.049721 | 6.378638 | 5.141944 | 11.610586 | 11.858898 |

Cross-validation performed better than baseline. Surprisingly, DMA performed best.