Smooth forecasting in R

Ivan Svetunkov

useR!

11th July 2019

Marketing Analytics and Forecasting

Lancaster University Management School

What is “smooth”?

Introduction

Forecasting Using State Space Models.

Implements Single Source of Error state space models (Snyder, 1985) for purposes of time series analysis and forecasting.

Motto of the package: give more flexibility to the user.

v2.5.1 on CRAN

But why?!

Let's go back in time... to October 21, 2015.

But why?!

I was doing my PhD...

But why?!

I was doing my PhD... with `ets()` from `forecast` package (Hyndman et al., 2019)...

...when I realised that I'm missing some features:

- Multiple steps ahead loss functions;
- Explanatory variables;
- More flexibility in the initialisation of the model;
- ...

What to do?

Develop your own package with exponential smoothing!

Introduction

Functions included in the package in 2019:

- Exponential smoothing in ETS framework, `es()`;
- Intermittent demand state space model, `es()`, `oes()`;
- State space ARIMA, `ssarima()`, `auto.ssarima()`;
- Multiple seasonal ARIMA, `msarima()`, `auto.msarima()`;
- Vector Exponential Smoothing, `ves()`;
- And others...

Not possible to cover everything, so let's have several case studies.

Introduction

Some posts about the features of the `es()` function (exerts from https://forecasting.svetunkov.ru):

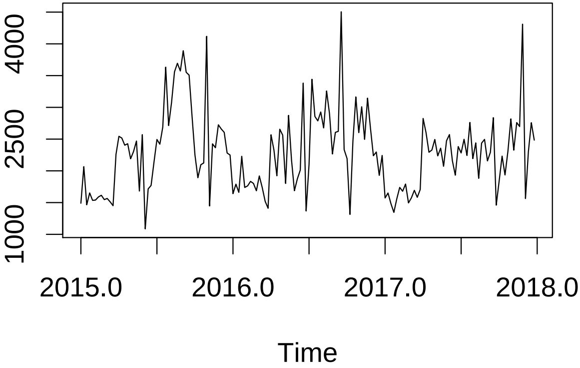
- Model types, model selection and combinations: http://tiny.cc/emxc9y, http://tiny.cc/znxc9y and http://tiny.cc/2oxc9y;
- Tuning the parameters of the model: http://tiny.cc/lqxc9y;
- Explanatory variables: http://tiny.cc/5uxc9y and http://tiny.cc/wwxc9y;
- Estimation of the model: http://tiny.cc/xsxc9y and http://tiny.cc/jtxc9y;
- Prediction intervals: http://tiny.cc/juxc9y;
- Intermittent demand: http://tiny.cc/w2xc9y.

Demand on fast moving products

Demand on fast moving products

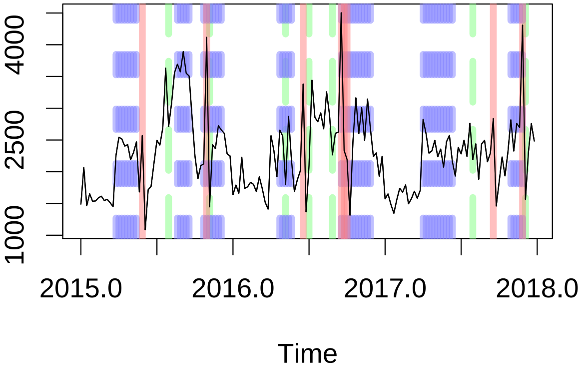
Fast moving products sales

Sales of beer...



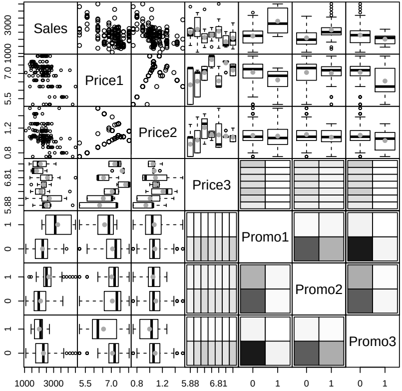
Fast moving products sales

With some promotions...



Fast moving products sales

And prices for the product and its competitors...



`spread()` function from `greybox`.

Fast moving products sales

We start with a seasonal exponential smoothing, ETS(MNM) model:

```
es(Sales, model="MNM", initial="backcasting",
   h=13, holdout=TRUE, interval="parametric", silent=FALSE)
```

Data is weekly, so estimating 52 seasonal indices might be difficult.

That's why we have `initial="backcasting"`.

Fast moving products sales

The output:

```
Time elapsed: 0.26 seconds
Model estimated: ETS(MNM)
Persistence vector g:
 alpha  gamma
0.2147 0.1366
Initial values were produced using backcasting.
Loss function type: MSE; Loss function value: 0.0273

Information criteria:
     AIC     AICc      BIC     BICc
2096.188 2096.361 2105.077 2105.505
```

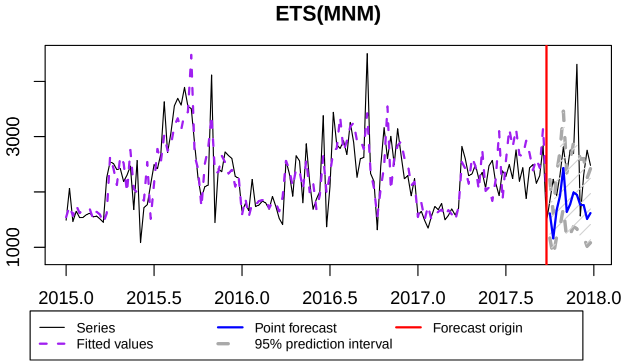
Fast moving products sales

...continued:

```
Error standard deviation: 0.1651
Sample size: 143
Number of estimated parameters: 3
Number of degrees of freedom: 140
95% parametric prediction interval were constructed
62% of values are in the prediction interval
Forecast errors:
MPE: 26.2%; sCE: -419.6%; Bias: 94.5%; MAPE: 28.2%
MASE: 1.792; sMAE: 33.6%; sMSE: 17.5%; RelMAE: 0.743; RelRMSE: 0.784
```

Fast moving products sales

The forecast...



...is not good.

Fast moving products sales

Let's introduce the explanatory variables:

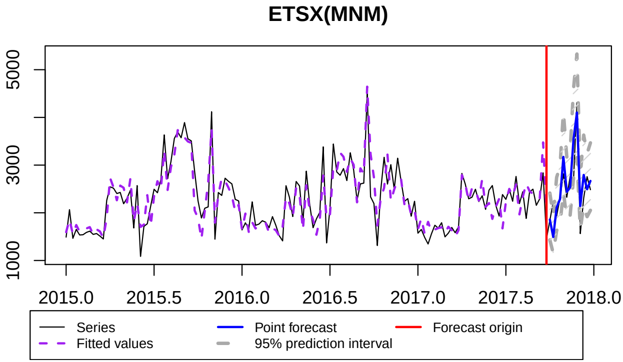
```
es(Sales, model="MNM", initial="backcasting", xreg=PromoData,
   h=13, holdout=TRUE, interval="parametric", silent=FALSE)
```

Fast moving products sales

The important chunks from the output:

```
Model estimated: ETSX(MNM)

Information criteria:
     AIC     AICc      BIC     BICc
2046.693 2048.046 2073.358 2076.717

77% of values are in the prediction interval

Forecast errors:
MPE: -5.2%; sCE: 57.3%; Bias: -33.9%; MAPE: 14%
MASE: 0.763; sMAE: 14.3%; sMSE: 3.3%; RelMAE: 0.316; RelRMSE: 0.34
```

Fast moving products sales



Much better now!

Fast moving products sales

Do variables selection inside the function, to remove redundant variables:

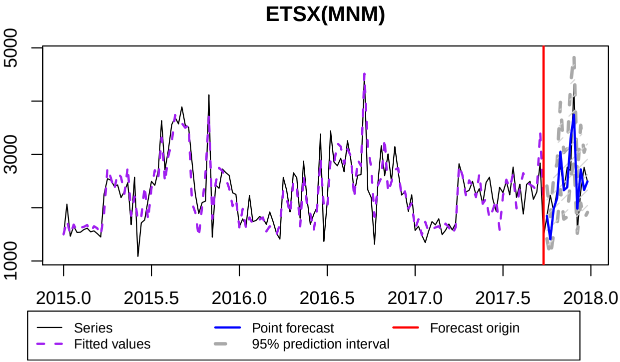
```
es(Sales, model="MNM", initial="backcasting", xreg=PromoData,
   xregDo="select",
   h=13, holdout=TRUE, interval="parametric", silent=FALSE)
```

Fast moving products sales

The important parts:

```
Information criteria:
     AIC     AICc      BIC     BICc
2047.095 2047.925 2067.835 2069.894

92% of values are in the prediction interval

Forecast errors:
MPE: 0.5%; sCE: -26.5%; Bias: 6.4%; MAPE: 12.6%
MASE: 0.733; sMAE: 13.8%; sMSE: 3.1%; RelMAE: 0.304; RelRMSE: 0.332
```

Fast moving products sales



Perfect!

Fast moving products sales

We could have done the same stuff automatically with `model="YYY"`:

```
es(Sales, model="YYY", initial="backcasting", xreg=PromoData,
    xregDo="select",
    h=13, holdout=TRUE, interval="parametric", silent=FALSE)
```

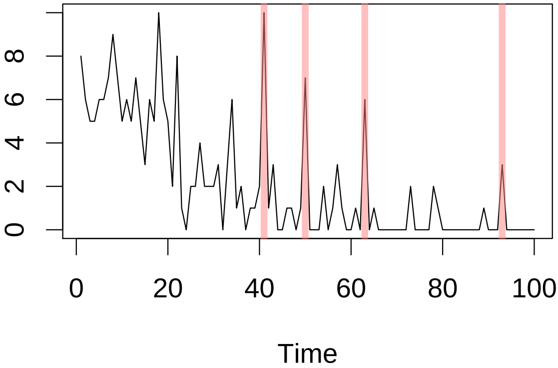
Potential improvement: include lead and lag effects of promotions.

Slow moving products sales

Demand on slow moving products

Slow moving products sales

Demand becoming obsolete + promotions:



Slow moving products sales

Start from smaller – Inverse odds ratio iETS model:

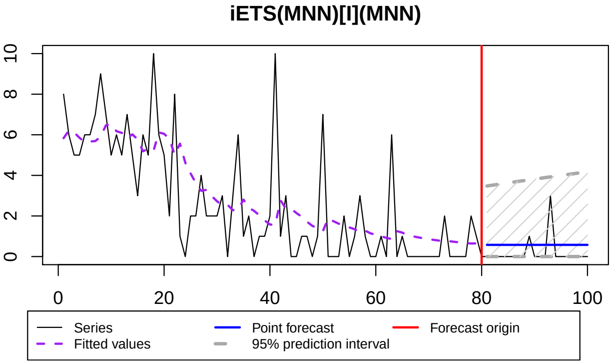
```
es(x, model="MNN", occurrence="inverse-odds-ratio",
   h=20, holdout=TRUE, interval="parametric", silent=FALSE)
```

Slow moving products sales

The important parts of the output:

```
Model estimated: iETS(MNN)
Occurrence model type: Inverse odds ratio

 alpha
0.2039

Information criteria:
     AIC     AICc      BIC     BICc
368.2378 368.5536 380.1479 372.0758

95% parametric prediction interval were constructed
100% of values are in the prediction interval

Forecast errors:
Bias: -79.6%; sMSE: 4%; RelRMSE: 1.097; sPIS: 2185%; sCE: 193.4%
```

Slow moving products sales



Slow moving products sales

Use trend and explanatory variables:

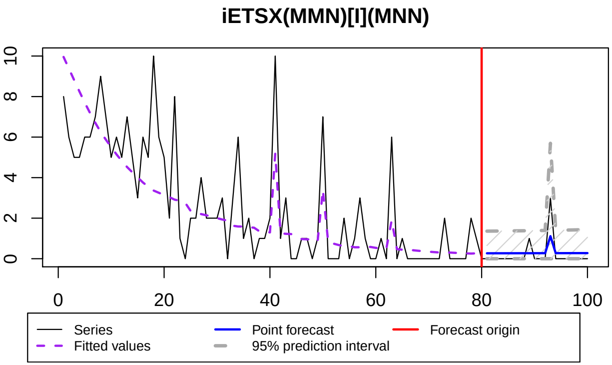
```
es(x, model="MMN", occurrence="inverse-odds-ratio", xreg=z,
   h=20, holdout=TRUE, interval="parametric", silent=FALSE)
```

Slow moving products sales

The important parts of the output:

```
Model estimated: iETSX(MMN)

 alpha   beta
0.0065 0.0065

Information criteria:
     AIC     AICc      BIC     BICc
313.5728 314.7235 332.6291 326.3862

Forecast errors:
Bias: -70.3%; sMSE: 1.8%; RelRMSE: 0.737; sPIS: 706.8%; sCE: 58.5%
```

Slow moving products sales



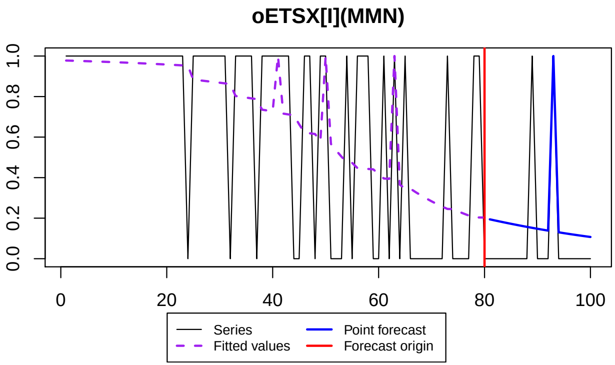
Slow moving products sales

Use oes() function in order to model the probability of occurrence.

Include multiplicative trend and the explanatory variable:

```
oesModel <- oes(x, model="MMN", occurrence="inverse-odds-ratio",
                xreg=z,
                h=20, holdout=TRUE, silent=FALSE)
```

Slow moving products sales



Slow moving products sales

Finally use it in the `es()`:

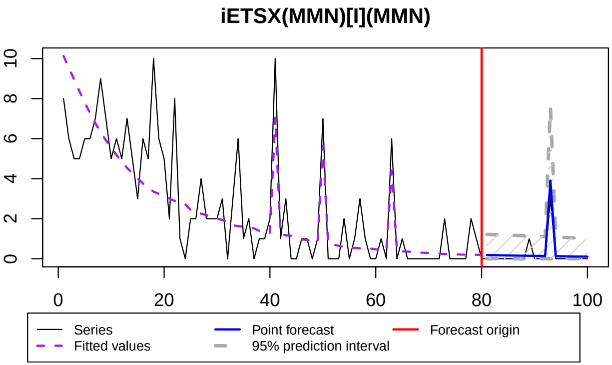
```
es(x, model="MMN", occurrence=oesModel, xreg=z,
   h=20, holdout=TRUE, interval="parametric", silent=FALSE)
```

Slow moving products sales

The important lines of the output:

```
 alpha  beta
0.0065 0.0065

Information criteria:
     AIC     AICc      BIC     BICc
303.3652 304.5159 317.6574 320.1786

Forecast errors:
Bias: -84.7%; sMSE: 0.6%; RelRMSE: 0.438; sPIS: 679.5%; sCE: 66.4%
```

Slow moving products sales



Slow moving products sales

Paper on iETS is under review at IJF.

Have a look at the working paper, if you want (Svetunkov and Boylan, 2017).

See `vignette("oes","smooth")` for more recent information.

Multiple seasonalities

Demand with multiple seasonalities

Multiple seasonalities

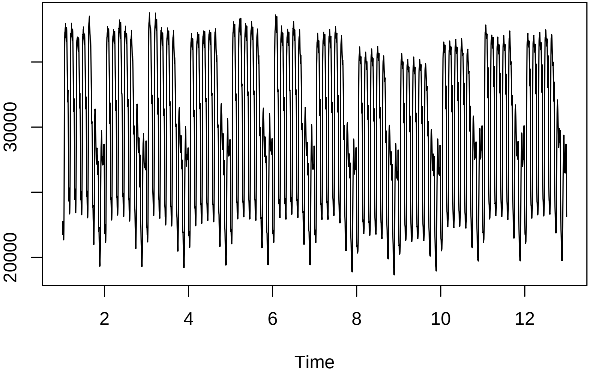
`msarima()` stands for "Multiple Seasonal ARIMA".

Flexibility of `msarima()`:

- Any orders you want, regulated by `order=list(ar=c(3,2,1), i=c(1,0,0), ma=c(1,2,3))`;
- Any lags you want, regulated by `lags=c(1,48,7*48)`.

Multiple seasonalities

Half-hourly electricity demand example (`taylor` from `forecast`).



Multiple seasonalities

Select the most suitable SARIMA model and produce forecasts:

```
auto.msarima(forecast::taylor,
             orders=list(ar=c(3,2,2),i=c(2,1,1),ma=c(3,2,2)),
             lags=c(1,48,48*7), h=48*7, holdout=TRUE,
             silent=FALSE)
```

Multiple seasonalities

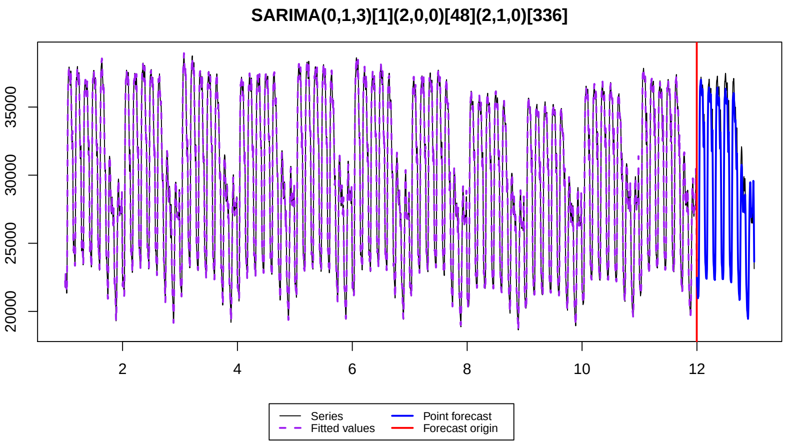
```
Time elapsed: 2125.07 seconds
Model estimated: SARIMA(0,1,3)[1](2,0,0)[48](2,1,0)[336]
Matrix of AR terms:
      Lag 48 Lag 336
AR(1)  0.394  -0.683
AR(2)  0.242  -0.403
Matrix of MA terms:
       Lag 1
MA(1)  0.062
MA(2) -0.041
MA(3) -0.073
Initial values were produced using backcasting.
8 parameters were estimated in the process
Residuals standard deviation: 147.774
Loss function type: MSE; Loss function value: 21837.251
```

Multiple seasonalities

...output continued...

```
Information criteria:
     AIC     AICc      BIC     BICc
47432.91 47432.95 47482.63 47482.79
Forecast errors:
MPE: 2.4%; sCE: -830.9%; Bias: 90.7%; MAPE: 2.7%
MASE: 1.254; sMAE: 2.8%; sMSE: 0.1%; RelMAE: 0.122; RelRMSE: 0.115
```

Multiple seasonalities



Multiple seasonalities

Alternatives from `smooth` to consider:

- Deterministic seasonality for half-hours (dummies);
- Deterministic seasonality for days of week;
- Do that with `es()`, `msarima()` or `gum()`;

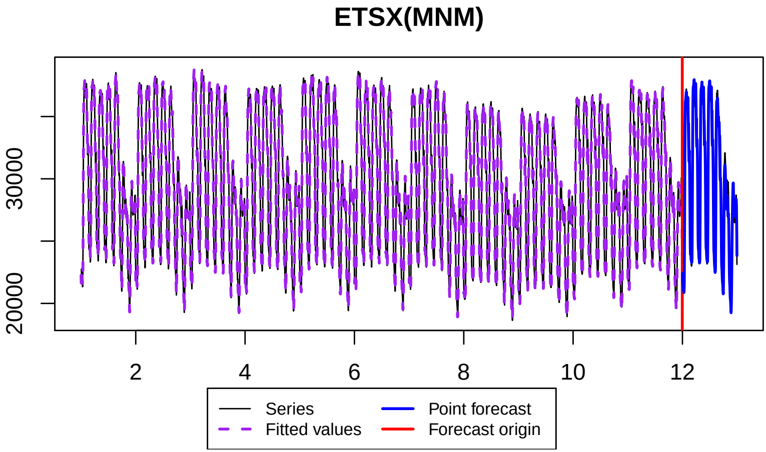
Multiple seasonalities

`es()` with deterministic daily seasonality.

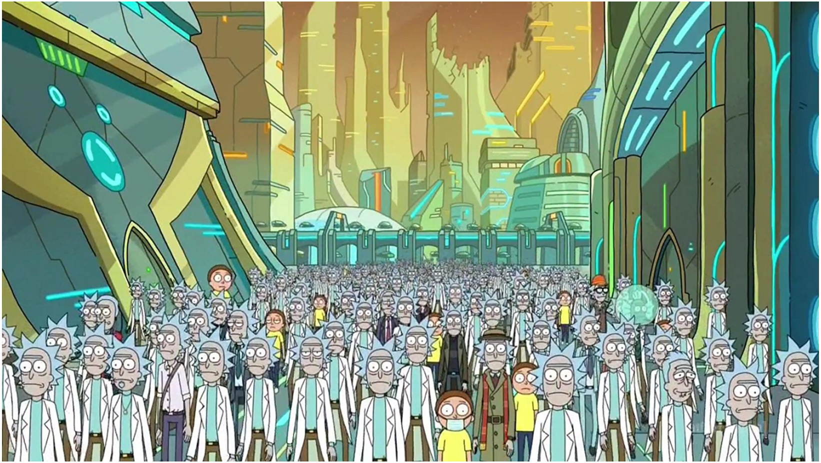
`taylorDummies` contains the dummies for days of week...

```
test <- es(taylor, model="MNM", xreg=taylorDummies,
          h=48*7, holdout=T, silent=F)
```

Multiple seasonalities



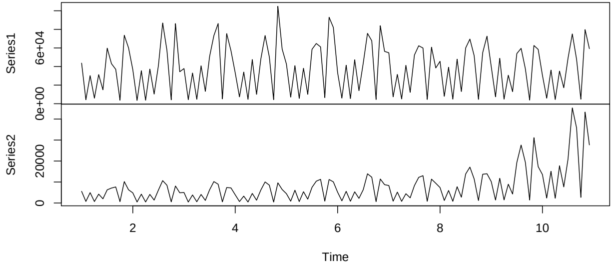
Multivariate data



Multivariate models

Multivariate data

Two products from the same category:



Multivariate data

They have similar seasonality;

They have similar number of components;

They might be related;

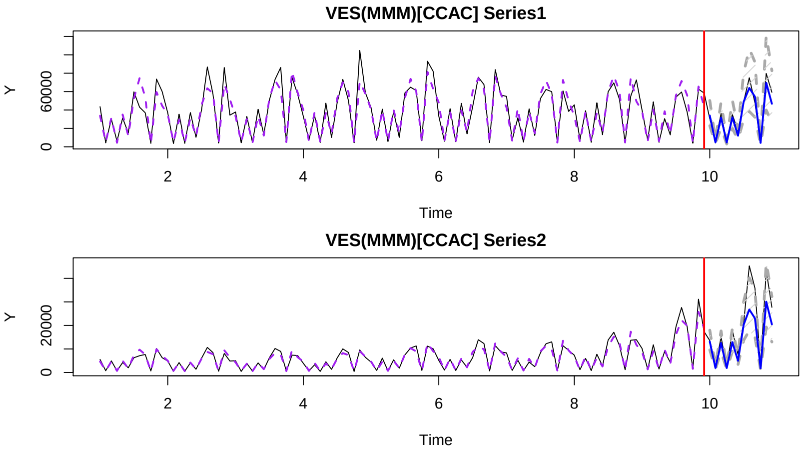
Use these features for forecasting...

Multivariate data

Apply vector exponential smoothing (`ves()` function, see vignettes):

```
ves(Y$data, model="MMM",
    persistence="common", initialSeason="common", seasonal="common",
    h=12, holdout=TRUE, silent=FALSE, interval="individual")
```

Multivariate data



Multivariate data

This is based on the research with Huijing Chen and John E. Boylan.

This was presented at ISF2019 by John E. Boylan.

What else?

What is left behind?

Some other functions and models implemented in the package:

- Complex exponential smoothing (Svetunkov and Kourentzes, 2018), `ces()` and `auto.ces()`;
- State space model constructor, `gum()`;
- Simple and centred moving averages in state space form (Svetunkov and Petropoulos, 2018): `sma()` and `cma()`;
- Simulation functions (ETS, ARIMA, VES, SMA, CES, GUM).

Finale

Thank you for your attention!

Ivan Svetunkov

i.svetunkov@lancaster.ac.uk

https://forecasting.svetunkov.ru

twitter: @iSvetunkov

Marketing Analytics and Forecasting | Lancaster University Management School

References I

Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., Yasmeen, F., 2019. forecast: Forecasting functions for time series and linear models. R package version 8.7. URL `http://pkg.robjhyndman.com/forecast`

Snyder, R. D., 1985. Recursive Estimation of Dynamic Linear Models. Journal of the Royal Statistical Society, Series B (Methodological) 47 (2), 272–276.

Svetunkov, I., Boylan, J. E., 2017. Multiplicative State-Space Models for Intermittent Time Series.

Svetunkov, I., Kourentzes, N., 2018. Complex Exponential Smoothing for Seasonal Time Series.

References II

Svetunkov, I., Petropoulos, F., sep 2018. Old dog, new tricks: a modelling view of simple moving averages. International Journal of Production Research 56 (18), 6034–6047.
URL https://www.tandfonline.com/doi/full/10.1080/00207543.2017.1380326