Smooth forecasting in R

Ivan Svetunkov

useR!

11th July 2019
What is “smooth”? 

smooth
Forecasting Using State Space Models.

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

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)...
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;
- ...

Ivan Svetunkov
Smooth forecasting in R
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?
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()``;
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()`. 
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...
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/emxc9y), [http://tiny.cc/znxc9y](http://tiny.cc/znxc9y) and [http://tiny.cc/2oxc9y](http://tiny.cc/2oxc9y);

- Tuning the parameters of the model: [http://tiny.cc/lqxc9y](http://tiny.cc/lqxc9y);

- Explanatory variables: [http://tiny.cc/5uxc9y](http://tiny.cc/5uxc9y) and [http://tiny.cc/wwxc9y](http://tiny.cc/wwxc9y);

- Estimation of the model: [http://tiny.cc/xsxc9y](http://tiny.cc/xsxc9y) and [http://tiny.cc/jtxc9y](http://tiny.cc/jtxc9y);

- Prediction intervals: [http://tiny.cc/juxc9y](http://tiny.cc/juxc9y);

- Intermittent demand: [http://tiny.cc/w2xc9y](http://tiny.cc/w2xc9y).
Demand on fast moving products
Fast moving products sales

Sales of beer...

<table>
<thead>
<tr>
<th>Time</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015.0</td>
<td>1000</td>
</tr>
<tr>
<td>2016.0</td>
<td>2500</td>
</tr>
<tr>
<td>2017.0</td>
<td>4000</td>
</tr>
<tr>
<td>2018.0</td>
<td></td>
</tr>
</tbody>
</table>

Ivan Svetunkov
Smooth forecasting in R
Fast moving products sales

With some promotions...

```
<table>
<thead>
<tr>
<th>Time</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015.0</td>
<td>1000</td>
</tr>
<tr>
<td>2016.0</td>
<td>2500</td>
</tr>
<tr>
<td>2017.0</td>
<td>4000</td>
</tr>
<tr>
<td>2018.0</td>
<td>6500</td>
</tr>
</tbody>
</table>
```
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:

```r
es(Sales, model="MNM", initial="backcasting",
    h=13, holdout=TRUE, interval="parametric", silent=FALSE)
```
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.
Let’s introduce the explanatory variables:

```r
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:

<table>
<thead>
<tr>
<th>AIC</th>
<th>AICc</th>
<th>BIC</th>
<th>BICc</th>
</tr>
</thead>
<tbody>
<tr>
<td>2046.693</td>
<td>2048.046</td>
<td>2073.358</td>
<td>2076.717</td>
</tr>
</tbody>
</table>

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

ETSX(MNM)

Much better now!
Fast moving products sales

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

```r
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:

<table>
<thead>
<tr>
<th>AIC</th>
<th>AICc</th>
<th>BIC</th>
<th>BICc</th>
</tr>
</thead>
<tbody>
<tr>
<td>2047.095</td>
<td>2047.925</td>
<td>2067.835</td>
<td>2069.894</td>
</tr>
</tbody>
</table>

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"`:

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

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

```r
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:

![Graph showing time series with sales peaks and demand becoming obsolete periods]

Ivan Svetunkov

Smooth forecasting in R
Slow moving products sales

Start from smaller – Inverse odds ratio iETS model:

```r
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:

<table>
<thead>
<tr>
<th>AIC</th>
<th>AICc</th>
<th>BIC</th>
<th>BICc</th>
</tr>
</thead>
<tbody>
<tr>
<td>368.2378</td>
<td>368.5536</td>
<td>380.1479</td>
<td>372.0758</td>
</tr>
</tbody>
</table>

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

\[ iETS(MNN)[I](MNN) \]
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)```

Ivan Svetunkov
Smooth forecasting in R
Slow moving products sales

The important parts of the output:

Model estimated: iETSX(MMN)

alpha beta
0.0065 0.0065

Information criteria:

<table>
<thead>
<tr>
<th>AIC</th>
<th>AICc</th>
<th>BIC</th>
<th>BICc</th>
</tr>
</thead>
<tbody>
<tr>
<td>313.5728</td>
<td>314.7235</td>
<td>332.6291</td>
<td>326.3862</td>
</tr>
</tbody>
</table>

Forecast errors:
Bias: -70.3%; sMSE: 1.8%; RelRMSE: 0.737; sPIS: 706.8%; sCE: 58.5%
Slow moving products sales

\[iETSX(MMN)[I](MNN)\]
Slow moving products sales

Use `oes()` function in order to model the probability of occurrence.
Use \texttt{oes()} function in order to model the probability of occurrence.

Include multiplicative trend and the explanatory variable:

\begin{verbatim}
 oesModel <- oes(x, model="MMN", occurrence="inverse-odds-ratio",
    xreg=z,
    h=20, holdout=TRUE, silent=FALSE)
\end{verbatim}
Slow moving products sales

\( oETSX[I](MMN) \)

- Series
- Fitted values
- Point forecast
- Forecast origin

Ivan Svetunkov
Smooth forecasting in R
Slow moving products sales

Finally use it in the `es()`:

```r
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:

\[
\begin{align*}
\text{alpha} & = 0.0065 \\
\text{beta} & = 0.0065 \\
\end{align*}
\]

Information criteria:

\[
\begin{align*}
\text{AIC} & = 303.3652 \\
\text{AICc} & = 304.5159 \\
\text{BIC} & = 317.6574 \\
\text{BICc} & = 320.1786 \\
\end{align*}
\]

Forecast errors:

\[
\begin{align*}
\text{Bias: } & -84.7\%; \quad \text{sMSE: } 0.6\%; \quad \text{RelRMSE: } 0.438\%; \quad \text{sPIS: } 679.5\%; \quad \text{sCE: } 66.4\%
\end{align*}
\]
Slow moving products sales

\[ \text{iETSX(MMN)[i](MMN)} \]

- Series
- Fitted values
- Point forecast
- 95% prediction interval
- Forecast origin

Ivan Svetunkov
Smooth forecasting in R
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).
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”.

Multiple seasonalities

\texttt{msarima()} stands for “Multiple Seasonal ARIMA”.

Flexibility of \texttt{msarima()}:

- Any orders you want, regulated by \texttt{order=list(ar=c(3,2,1), i=c(1,0,0), ma=c(1,2,3))};
- Any lags you want, regulated by \texttt{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:

```r
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:

<table>
<thead>
<tr>
<th>Lag 48</th>
<th>Lag 336</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR(1) 0.394</td>
<td>-0.683</td>
</tr>
<tr>
<td>AR(2) 0.242</td>
<td>-0.403</td>
</tr>
</tbody>
</table>

Matrix of MA terms:

<table>
<thead>
<tr>
<th>Lag 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA(1) 0.062</td>
</tr>
<tr>
<td>MA(2) -0.041</td>
</tr>
<tr>
<td>MA(3) -0.073</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Information criteria</th>
<th>AIC</th>
<th>AICc</th>
<th>BIC</th>
<th>BICc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>47432.91</td>
<td>47432.95</td>
<td>47482.63</td>
<td>47482.79</td>
</tr>
</tbody>
</table>

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

SARIMA(0,1,3)[1](2,0,0)[48](2,1,0)[336]

Series
Fitted values
Point forecast
Forecast origin

Ivan Svetunkov
Smooth forecasting in R
Multiple seasonalities

Alternatives from smooth to consider:

- Deterministic seasonality for half-hours (dummies);
- Deterministic seasonality for days of week;
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

\texttt{es()} with deterministic daily seasonality.

taylorDummies contains the dummies for days of week...

test <- \texttt{es(taylor, model="MNM", xreg=taylorDummies, h=48*7, holdout=T, silent=F)}
Multiple seasonalities

ETSX(MNM)

- Series
- Fitted values
- Point forecast
- Forecast origin

Ivan Svetunkov
Smooth forecasting in R
Multivariate data

Multivariate models
Multivariate data

Two products from the same category:
Multivariate data

Apply vector exponential smoothing \((\text{ves}())\) function, see vignettes):

\[
\text{ves}(\text{Y$data, model="MMM", persistence="common", initialSeason="common", seasonal="common", h=12, holdout=TRUE, silent=FALSE, interval="individual"})
\]
Multivariate data

**VES(MMM)[CCAC] Series1**

![Graph of VES(MMM)[CCAC] Series1](image1)

**VES(MMM)[CCAC] Series2**

![Graph of VES(MMM)[CCAC] Series2](image2)
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()`;
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()`;
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()`;
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

That's all Folks!
Thank you for your attention!

Ivan Svetunkov
i.svetunkov@lancaster.ac.uk
https://forecasting.svetunkov.ru
twitter: @iSvetunkov
References I


References II