Seventh International Conference on System Modelling & Advancement on Research Trends. Moradabad, India.

November 23-24, 2018

Data Schemas for Forecasting (with examples in R)

Sai, C., Davydenko, A., & Shcherbakov, M.

to handle forecast data efficiently?

What data structures do we need to use in order

Forecast evaluation framework

Our aim:

to find appropriate **data structures** that can be used as a base for implementing a general forecast evaluation framework.

The framework should allow these capabilities:

- 1) forecast data storage and exchange,
- 2) exploratory analysis of forecasts and time series
- 3) measuring forecasting performance.

Requirements:

the **data structures** should be <u>simple</u>, but <u>cross-platform</u>, <u>flexible</u>, and <u>sufficient</u> to implement the above capabilities.

Setup

- 1) We have a set of time series, the set can contain from 1 to millions of series.
- 2) For each series we want to store and update actuals and (numeric) forecasts
- 3) We want to store and update out-of-sample forecasts, made with alternative methods at different origins with different horizons. Calculating forecasts may take relatively long time.
- 4) We may want to store not only point forecasts, but prediction intervals, density forecasts and any additional information

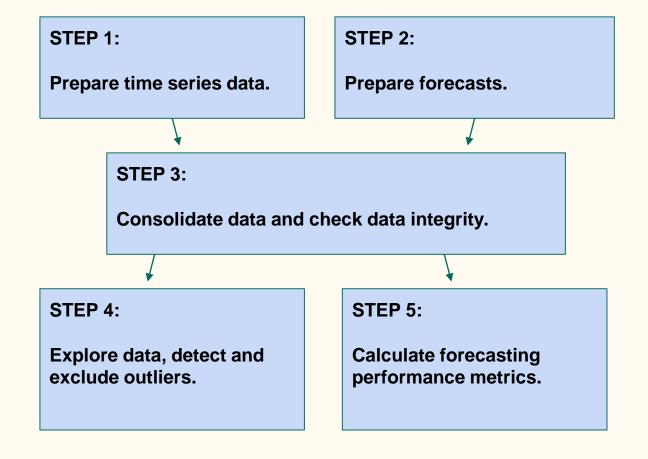
Given the above settings, we want to explore forecasts and to evaluate forecasting performance regularly.

Target audience

This presentation particularly targeted at you if you are

- a researcher wanting to conduct forecasting competitions
- a researcher wanting to compare a new forecasting methods against alternatives
- a practitioner wanting to develop software components to evaluate forecasting performance

Our approach: The general framework



Step 1: Prepare time series data

In order to store time series data, we propose the following schema:

Time Series Table Schema (TSTS):

| Field name (column name) | Description | Examples |
|--------------------------------|--|--|
| *series_id | Time series identifier - a unique name that identifies a time series | "Y1" |
| *timestamp | Any representation of the period to which the observation relates. | "01.01.1997" in case of daily data, "Sep 1997" in case of monthly data, "Week 49, 1997" in case of weekly data |
| value | The value observed | 1000 |

Example:

| series_id | value | timestamp |
|-----------|---------|-----------|
| <u>Y1</u> | 3103.96 | 1984 |
| Y1 | 3360.27 | 1985 |
| Y1 | 3807.63 | 1986 |
| Y1 | 4387.88 | 1987 |
| Y1 | 4936.99 | 1988 |
| Y1 | 5379.75 | 1989 |
| Y1 | 6158.68 | 1990 |
| Y1 | 6876.58 | 1991 |
| Y2 | 5389.80 | 1984 |
| Y2 | 5384.40 | 1985 |
| | | |

Step 2: Prepare forecasts

In order to store forecasts, we propose the following schema:

Forecasts Table Schema (FTS):

| *series_id | *timestamp | *origin_timestamp | *horizon | *method | forecast | Lo95 | Hi95 |
|------------|------------|-------------------|----------|---------|----------|------|------|
| | | | | | | | |

Example:

| method | timestamp | origin_timestamp | forecast | horizon | Lo90 | Hi90 |
|--------------|-----------------|---|---|---|---|---|
| A | 1989 | 1988 | 5406.43 | 1 | 5183.349 | 5629.511 |
| A | 1990 | 1988 | 5875.96 | 2 | 5652.879 | 6099.041 |
| A | 1991 | 1988 | 6345.48 | 3 | 6122.399 | 6568.561 |
| В | 1989 | 1988 | 5473.87 | 1 | 5250.789 | 5696.951 |
| В | 1990 | 1988 | 6010.43 | 2 | 5787.349 | 6233.511 |
| В | 1991 | 1988 | 6546.63 | 3 | 6323.549 | 6769.711 |
| \mathbf{C} | 1989 | 1988 | 5406.43 | 1 | 5183.349 | 5629.511 |
| \mathbf{C} | 1990 | 1988 | 5875.96 | 2 | 5652.879 | 6099.041 |
| \mathbf{C} | 1991 | 1988 | 6345.48 | 3 | 6122.399 | 6568.561 |
| A | 1989 | 1988 | 4142.60 | 1 | 3919.519 | 4365.681 |
| A | 1990 | 1988 | 4055.47 | 2 | 3832.389 | 4278.551 |
| | A A B B C C C A | A 1989 A 1990 A 1991 B 1989 B 1990 B 1991 C 1989 C 1990 C 1991 A 1989 | A 1989 1988 A 1990 1988 A 1991 1988 B 1989 1988 B 1990 1988 B 1991 1988 C 1989 1988 C 1990 1988 C 1990 1988 C 1990 1988 C 1990 1988 A 1989 1988 | A 1989 1988 5406.43 A 1990 1988 5875.96 A 1991 1988 6345.48 B 1989 1988 5473.87 B 1990 1988 6010.43 B 1991 1988 6546.63 C 1989 1988 5406.43 C 1990 1988 5875.96 C 1991 1988 6345.48 A 1989 1988 4142.60 | A 1989 1988 5406.43 1 A 1990 1988 5875.96 2 A 1991 1988 6345.48 3 B 1989 1988 5473.87 1 B 1990 1988 6010.43 2 B 1991 1988 6546.63 3 C 1989 1988 5406.43 1 C 1990 1988 5875.96 2 C 1991 1988 6345.48 3 A 1989 1988 4142.60 1 | A 1989 1988 5406.43 1 5183.349 A 1990 1988 5875.96 2 5652.879 A 1991 1988 6345.48 3 6122.399 B 1989 1988 5473.87 1 5250.789 B 1990 1988 6010.43 2 5787.349 B 1991 1988 6546.63 3 6323.549 C 1989 1988 5406.43 1 5183.349 C 1990 1988 5875.96 2 5652.879 C 1991 1988 6345.48 3 6122.399 A 1989 1988 4142.60 1 3919.519 |

Step 3: Consolidate data

Here we need to obtain a table containing both actuals and forecasts, this will be needed to implement some elements for exploratory analysis and accuracy measurement.

In order to obtain the consolidated data set we propose to use the Actuals and Forecasts Table Schema (AFTS).

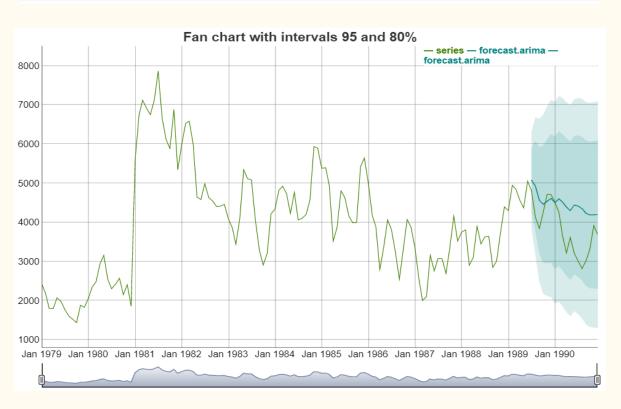
This schema is the same as the **FTS** schema, but additional column "value" is used to represent the actual value of time series.

Example:

| series_id | value | method | timestamp | $origin_timestamp$ | forecast | horizon | Lo90 | Hi90 |
|-----------|---------|-----------------|-----------|---------------------|----------|---------|----------|----------|
| Y1 | 5379.75 | A | 1989 | 1988 | 5406.43 | 1 | 5183.349 | 5629.511 |
| Y1 | 6158.68 | A | 1990 | 1988 | 5875.96 | 2 | 5652.879 | 6099.041 |
| Y1 | 6876.58 | A | 1991 | 1988 | 6345.48 | 3 | 6122.399 | 6568.561 |
| Y1 | 5379.75 | В | 1989 | 1988 | 5473.87 | 1 | 5250.789 | 5696.951 |
| Y1 | 6158.68 | В | 1990 | 1988 | 6010.43 | 2 | 5787.349 | 6233.511 |
| Y1 | 6876.58 | В | 1991 | 1988 | 6546.63 | 3 | 6323.549 | 6769.711 |
| Y1 | 5379.75 | \mathbf{C} | 1989 | 1988 | 5406.43 | 1 | 5183.349 | 5629.511 |
| Y1 | 6158.68 | $^{\mathrm{C}}$ | 1990 | 1988 | 5875.96 | 2 | 5652.879 | 6099.041 |
| Y1 | 6876.58 | $^{\mathrm{C}}$ | 1991 | 1988 | 6345.48 | 3 | 6122.399 | 6568.561 |
| Y2 | 4793.20 | A | 1989 | 1988 | 4142.60 | 1 | 3919.519 | 4365.681 |

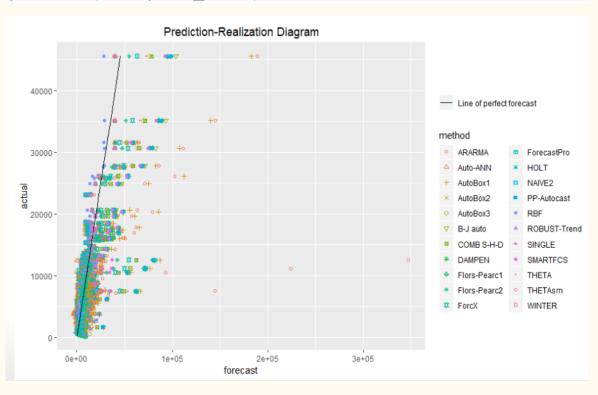
Step 4: Exploratory analysis

plotSeries(example1_afts, seried_id = "M1")



Step 4: Exploratory analysis

plotPRD(example1_afts)

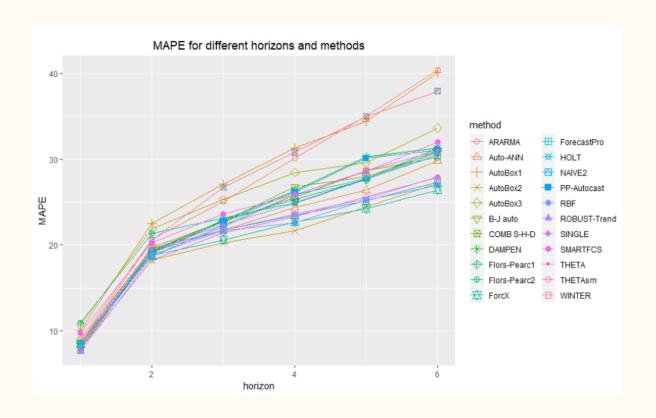


Step 5: Measuring forecasting performance

calculateMAPEs(example1_afts)

| | horizon = 1 | horizon = 2 | horizon = 3 | horizon = 4 | horizon = 5 | horizon = 6 |
|--------------|-------------|-------------|-------------|-------------|-------------|-------------|
| NATVE2 | 8.360053 | 19.23712 | 21.70531 | 23.45871 | 25.17578 | 27.35164 |
| | | | | | | |
| SINGLE | 8.426719 | 19.53460 | 21.70985 | 23.59725 | 25.35748 | 27.93413 |
| HOLT | 8.504891 | 20.57738 | 26.74072 | 30.80756 | 34.94463 | 37.94606 |
| DAMPEN | 8.161127 | 19.23165 | 22.88949 | 26.32286 | 30.25410 | 31.27435 |
| WINTER | 8.504891 | 20.57738 | 26.74072 | 30.80756 | 34.94463 | 37.94606 |
| COMB S-H-D | 7.964892 | 19.02728 | 22.76000 | 25.56244 | 28.63649 | 30.24861 |
| B-J auto | 8.638050 | 19.71086 | 22.78263 | 26.77603 | 27.99026 | 30.82170 |
| AutoBoxl | 10.119198 | 22.51186 | 27.07629 | 31.31042 | 34.37756 | 40.08493 |
| AutoBox2 | 7.951192 | 18.21996 | 20.24227 | 21.65581 | 24.46921 | 27.17624 |
| AutoBox3 | 10.698830 | 21.89010 | 25.29647 | 28.45540 | 29.57899 | 33.62135 |
| ROBUST-Trend | 7.606495 | 18.64720 | 22.39440 | 24.83567 | 27.61491 | 30.66538 |
| ARARMA | 9.091266 | 20.68177 | 25.10429 | 30.14883 | 34.99774 | 40.38033 |
| Auto-ANN | 8.956602 | 19.67521 | 21.76107 | 24.36152 | 26.41399 | 29.81788 |
| Flors-Pearcl | 8.561016 | 19.38149 | 22.80052 | 25.34184 | 27.62398 | 30.95579 |
| Flors-Pearc2 | 10.903332 | 21.38609 | 23.17941 | 24.91399 | 27.72512 | 31.29920 |
| PP-Autocast | 8.141452 | 19.19054 | 22.75382 | 26.17481 | 30.09973 | 31.09496 |
| ForecastPro | 8.426093 | 18.77205 | 22.10483 | 25.87735 | 27.74920 | 30.45980 |
| SMARTFCS | 9.796722 | 20.29223 | 23.64564 | 25.85210 | 28.55908 | 31.99116 |

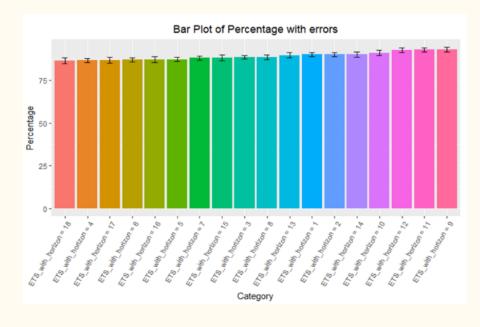
Step 5: Measuring forecasting performance



Step 5: Measuring forecasting performance

validatePIs(example2_afts)

```
category cases total percentage
                                                 Lo 95 Hi 95
18 ETS with horizon = 18
                          1238
                                1428
                                         86.695 84.823 88.414
    ETS with horizon = 4
                          2604
                                3003
                                         86.713 85.447 87.908
17 ETS with horizon = 17
                                1428
                                         86.905 85.044 88.612
                          1241
    ETS with horizon = 6
                          2615
                                3003
                                          87.080 85.827 88.259
16 ETS with horizon = 16
                          1247
                                1428
                                          87.325 85.487 89.007
    ETS with horizon = 5
                          2623
                                3003
                                          87.346 86.104 88.515
    ETS with horizon = 7
                          2078
                                2358
                                          88.126 86.751 89.404
15 ETS with horizon = 15
                          1261
                                1428
                                          88.305 86.524 89.927
   ETS with horizon = 3
                          2661
                                3003
                                          88.611 87.421 89.726
   ETS with horizon = 8
                          2091
                                2358
                                          88.677 87.328 89.928
13 ETS with horizon = 13
                                1428
                          1282
                                          89.776 88.087 91.299
    ETS_with_horizon = 1
                          2709
                                3003
                                          90.210 89.091 91.250
    ETS with horizon = 2
                          2709
                                3003
                                          90.210 89.091 91.250
14 ETS with horizon = 14
                          1290
                                1428
                                          90.336 88.685 91.819
10 ETS with horizon = 10
                          1302
                                1428
                                          91.176 89.584 92.597
12 ETS with horizon = 12
                          1325
                                1428
                                          92.787 91.320 94.075
11 ETS with horizon = 11
                          1328
                                1428
                                          92.997 91.548 94.266
    ETS with horizon = 9
                                1428
                                          93.137 91.700 94.394
```



Conclusions

The approach proposed allows:

- to represent your forecast data in the form suitable for further analysis of forecasting performance and for the exchange with other researchers
- to explore forecasts in order to make sure that your data is correct and ready for accuracy evaluation
- to evaluate forecasting performance based on the data formats defined

Our approach does not depend on any platform or programming language, it just defines the general methodology for handling forecast data.

For more info on this project, please visit:

<u>forvis.github.io</u>