Technical Report Series, N. 1, August 2016

Department of Statistical Sciences
University of Padua
Italy

Life-cycle forecasting for pharmaceutical drugs. A structured approach with Mathematica

Cinzia Mortarino
Department of Statistical Sciences
University of Padua
Italy

Abstract: The study of the lifecycle of existing pharmaceutical products within a common therapeutic category may provide some insight into the expected performance of a new drug with comparable efficacy from the medical point of view. Whenever the category is crowded, however, this approach requires the analysis of hundreds, or even thousands, of sales' time series. The aim of this report is to present a technical tool to deal with data inspection, univariate diffusion model fitting, and forecasting storage for a large number of products. This is realized with the software Mathematica (Versions 9 or 10).

Keywords: nonlinear model fitting, diffusion process, generalized Bass model, intervention, dynamic market potential

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI SCIENZE STATISTICHE

Contents

1 Introduction **1**

2 Input & Output Data Structure **2**

3 Model fitting **5**
3.1 Cumulative data 5
3.2 Instantaneous data 10

4 A not-so-rare estimation issue **15**

5 Final remarks **18**

Appendix A. Innovation diffusion models and statistical inference **20**

Appendix B. Hints about Mathematica's commands sintax **22**

Department of Statistical Sciences
Via Cesare Battisti, 241
35121 Padova
Italy

tel: +39 049 8274168
fax: +39 049 8274170
http://www.stat.unipd.it

Corresponding author:
Cinzia Mortarino
tel: +39 049 827 4184
mortarino@stat.unipd.it
http://www.stat.unipd.it/~mortarino

Life-cycle forecasting for pharmaceutical drugs. A structured approach with Mathematica

Cinzia Mortarino
Department of Statistical Sciences
University of Padua
Italy

Abstract: The study of the lifecycle of existing pharmaceutical products within a common therapeutic category may provide some insight into the expected performance of a new drug with comparable efficacy from the medical point of view. Whenever the category is crowded, however, this approach requires the analysis of hundreds, or even thousands, of sales' time series. The aim of this report is to present a technical tool to deal with data inspection, univariate diffusion model fitting, and forecasting storage for a large number of products. This is realized with the software Mathematica (Versions 9 or 10).

Keywords: nonlinear model fitting, diffusion process, generalized Bass model, intervention, dynamic market potential

1 Introduction

The entry into the market of a new pharmaceutical drug requires in general an ex-ante assessment of its performance in terms of package sales. This is relevant both for pharmaceutical companies but also for payers negotiating price and market access conditions.

This assessment, which has to be done before the market launch when data cannot be available, is often performed through qualitative methods based on management or potential customers judgements (Goodwin et al., 2014). Quantitative methods based on the characteristics of similar products already launched in the same category may provide an interesting completion to judgemental analysis (Goodwin et al., 2013; Kim et al., 2013; Lee et al., 2014; Wright and Stern, 2015).

In Guseo et al. (2016), a new approach is proposed to exploit the dynamic features of the therapeutic category evolving both under the pressure of a patients' population growth and due to the presence of drugs based on more recent active compounds. In addition, the market composition changes for the launch of different packages/dosages of an already commercialized compound or for the availability of generic packages for out-of-patent products. In Jommi et al. (2016), this dynamic approach is used to provide estimates for the future sales (number of packages) of a prioritized drug and its emerging competitors within the antidiabetic category (ATC A10B) in order to perform a robust Budget Impact Analysis.

The first step for the application of this approach requires the description of

the time series of monthly or quarterly packages sold for each specialty. A specialty here represents a single IMA (International Marketing Authorization) code, pertaining to the same drug, with the same dosage and package. The choice to work at the lowest level, the specialty level (without aggregating data at the product or at the active compound level), is motivated by the need to link each specialty to a single time point (its launch time into the market, that is its birth date) and also by the consideration that aggregation of different lifecycles usually hides the evolutionary performances of single components, thus weakening the prerequisites for forecasting. The description of single mean sales trajectories is performed through the Guseo Guidolin Model (GGM), Guseo and Guidolin (2009), which, in spite of a reduced number of parameters, proved to be very flexible to adapt to different sales' patterns (see Appendix A for a short summary of univariate diffusion models and the references there cited for a more detailed study).

The model fitting of the GGM (possibly with appropriate intervention functions) requires a nonlinear estimation procedure (nonlinear least squares, NLS), which has to be repeated, with good initial values, for all the specialties' time series. After that step, parameter estimates and forecasted future sales values have to be efficiently stored. This is not a complex task, but the huge number of parallel specialties, with different starting points and length, may introduce some organizational challenges. In this paper, it is proposed the detailed description of a Mathematica notebook (Versions 9 or 10) built to face all these steps.

The choice to use Mathematica is grounded on its superiority in the implementation of algorithms for nonlinear model fitting together with its efficiency in managing large datasets with repeated sets of commands.

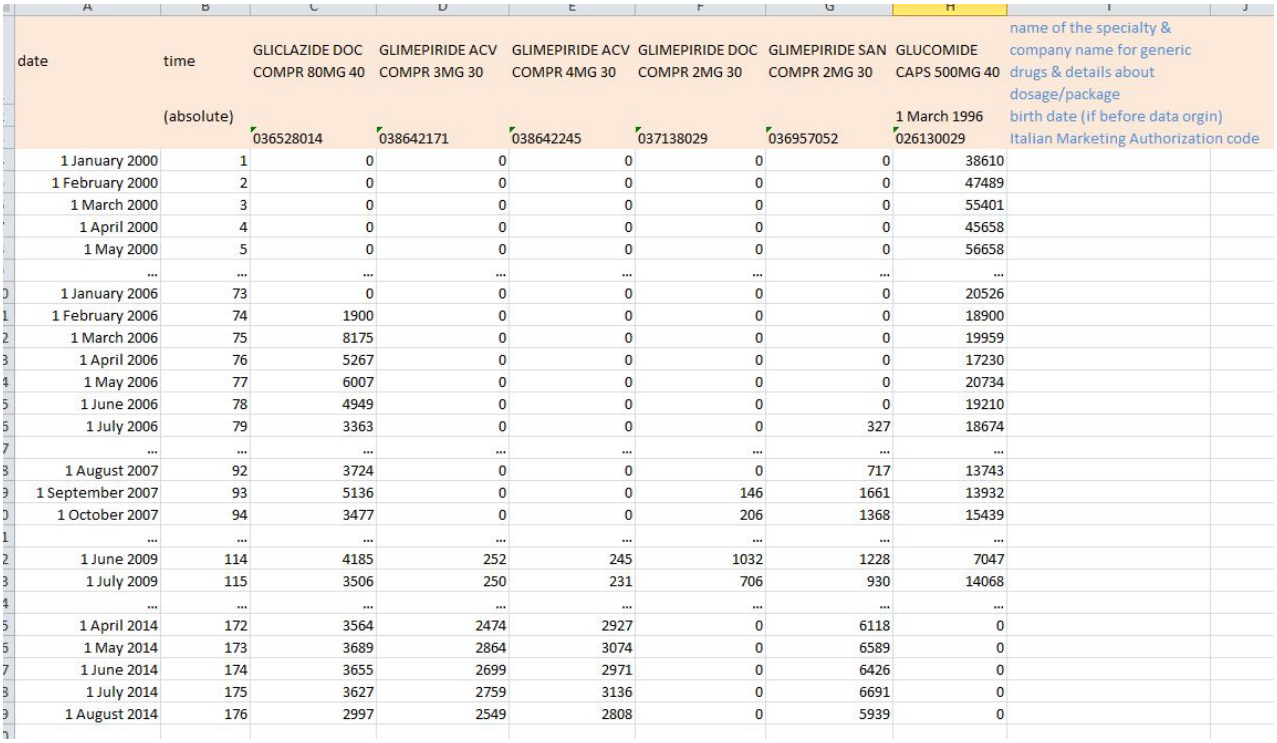
In Section 2, input format and output of the Mathematica notebook are described. Section 3 describes the model fitting procedure, separately for cumulative and instantaneous data. In Section 4, a suggestion to improve the set of initial values when some parameters are out of range is given. Some concluding remarks are given in Section 5. Appendix A summarizes pertinent innovation diffusion models. Appendix B contains some hints about the syntax of Mathematica's commands for readers not familiar with this software.

2 Input & Output Data Structure

The notebook starts by taking as input a csv format or a xls (xlsx) file (but also many other formats are accepted by Mathematica), with specialty instantaneous sales time series as columns (see Figure 1). For an application to Italy, we utilized monthly sales data starting from January 2000 until August 2014 (Source: IMS Health, Italy).

The longer is the period covered by data, the easier is to describe a precise picture of the past evolution of the therapeutic category. It is important to know sales data since the launch of the specialty into the country market for as many specialties as possible. But for older specialties, launched before the data starting point, we can analyze a truncated time series, provided a reasonably reliable information about their birth dates is available (here this information is recorded in the second

Figure 1: Example of input data format (Excel sheet).



	A	B	C	D	E	F	G	H	I	J
	date	time	GLICLAZIDE DOC COMPR 80MG 40	GLIMEPIRIDE ACV COMPR 3MG 30	GLIMEPIRIDE ACV COMPR 4MG 30	GLIMEPIRIDE DOC COMPR 2MG 30	GLIMEPIRIDE SAN COMPR 2MG 30	GLUCOMIDE CAPS 500MG 40	name of the specialty & company name for generic drugs & details about dosage/package
		(absolute)	036528014	038642171	038642245	037138029	036957052	1 March 1996 026130029	birth date (if before data orgin) Italian Marketing Authorization code
	1 January 2000	1	0	0	0	0	0	38610	
	1 February 2000	2	0	0	0	0	0	47489	
	1 March 2000	3	0	0	0	0	0	55401	
	1 April 2000	4	0	0	0	0	0	45658	
	1 May 2000	5	0	0	0	0	0	56658	
	...	...	...	...	...	...	...	...	
0	1 January 2006	73	0	0	0	0	0	20526	
1	1 February 2006	74	1900	0	0	0	0	18900	
2	1 March 2006	75	8175	0	0	0	0	19959	
3	1 April 2006	76	5267	0	0	0	0	17230	
4	1 May 2006	77	6007	0	0	0	0	20734	
5	1 June 2006	78	4949	0	0	0	0	19210	
6	1 July 2006	79	3363	0	0	0	327	18674	
7	...	...	...	...	...	...	...	...	
8	1 August 2007	92	3724	0	0	0	717	13743	
9	1 September 2007	93	5136	0	0	146	1661	13932	
0	1 October 2007	94	3477	0	0	206	1368	15439	
1	...	...	...	...	...	...	...	...	
2	1 June 2009	114	4185	252	245	1032	1228	7047	
3	1 July 2009	115	3506	250	231	706	930	14068	
4	...	...	...	...	...	...	...	...	
5	1 April 2014	172	3564	2474	2927	0	6118	0	
6	1 May 2014	173	3689	2864	3074	0	6589	0	
7	1 June 2014	174	3655	2699	2971	0	6426	0	
8	1 July 2014	175	3627	2759	3136	0	6691	0	
9	1 August 2014	176	2997	2549	2808	0	5939	0	

line of the file). The first two columns of the file contain a time reference, that is the month and year (first column) and an absolute time index (second column) starting with the data origin (in our example, $t = 1$ in January 2000).

```
data = Import["C:\\PATH\\example.xls"];
data = data[[1]];
Dimensions[data]
   Out:  {179,8}
ncol = 8; nrow = 179;
THIS COMMAND CONVERTS AIC TO A TEXT STRING TO AVOID THE SCIENTIFIC
NOTATION
Table[data[[3, h]] = ToString[Round[data[[3, h]], 1]], {h, 3, ncol}];
```

Table 1 contains some notation. Before proceeding with the analysis of single specialties, the structure for estimates and forecasts storage has to be defined. `Pr` represents the final point for the forecasting period (t=230, February 2019). The matrix named `Output` will contain the forecasted values for each times series. The parameter estimated values for the main set of 5 parameters of the GGM (M, p_c, q_c, p, q) will be stored in `ParameterEstimates,` together with the birth date of the specialty.

Table 1: Notation.

`ncol`	number of columns in the Excel sheet (equals the number of specialties to analyze + 2)
`nrow`	number of rows in the Excel sheet (equals the number of months covered at maximum by data + 3)
`Pr`	final point of the forecasting period (t=230, corresponding to February 2019)
`arg`	vector of values from 1 to `Pr` used for forecasts evaluation
`Output`	matrix to store forecasts for each specialty (same structure as the data matrix, with 54 extra rows corresponding to months where no data have been observed but forecasts are required)
`ParameterEstimates`	matrix to store the launch time and the parameter estimates for (M, p_c, q_c, p, q) for each specialty to be analyzed
`t_init`	launch time for the specialty under analysis
`len, Nd`	number of available data
`label`	string to be used as identifier for the specialty under analysis in the corresponding plots
`TOT`	data matrix for the specialty under analysis to be used in `NonlinearRegression`
`serieTOT`	column in the data matrix corresponding to the specialty under analysis
`aT`	vector with instantaneous fitted data (in–sample) for the specialty under analysis to be used for squared correlation coefficient's evaluation
`bT`	vector with instantaneous sales data for the specialty under analysis to be used for plotting and for squared correlation coefficient's evaluation
`decrem`	time shift between the launch time and the data origin in the case of density estimation (number of months)
`Prtemp`	extension of `Pr` in the case of density estimation (`Pr`+`decrem`)

```
 Pr = 230;
arg = Table[i, {i, 1, Pr, 1}];
MATRIX NAMED OUTPUT WILL CONTAIN THE FORECASTED VALUES FOR EACH TIMES SERIES.
THIS INITIALIZATION PREPARES COLUMNS' NAMES AND AIC AND TIME REFERENCES ACCORDING
TO THE INPUT MATRIX NAMED DATA (ADDING ALSO DATES TO COVER THE FORECASTING PERIOD)
Output = Table[0, {i, 1, Pr + 3}, {j, 1, ncol}];
Output[[1, All]] = data[[1, All]]; Output[[2, All]] = data[[2, All]];
Output[[3, All]] = data[[3, All]];
Table[Output[[j, 2]] = j - 3, {j, 4, Pr + 3}];
Table[Output[[j, 1]] = data[[j, 1]], {j, 4, nrow}];
extend = DateRange[DateObject[2014, 9, 1, TimeObject[0, 0, 0.‘, TimeZone -> 2.‘],
   TimeZone -> 2.‘], DateObject[2019, 2, 1, TimeObject[0, 0, 0.‘, TimeZone -> 2.‘],
   TimeZone -> 2.‘], "Month"];
Table[Output[[h, 1]] = extend[[h - nrow, 1]], h, nrow + 1, Pr + 3];
MATRIX NAMED PARAMETERESTIMATES WILL STORE THE PARAMETER ESTIMATED VALUES
ParameterEstimates = Table[0, {i, 1, ncol}, {j, 1, 7}];
ParameterEstimates[[1, 1]] = "name";
ParameterEstimates[[1, 2]] = "t_init";
ParameterEstimates[[1, 3]] = "M";
⋮
ParameterEstimates[[1, 7]] = "q";
Table[ParameterEstimates[[2, h]] = "", {h, 1, 7}];
```

At that step, matrix `Output` only contains the structure for predicted values (see Figure 2), while matrix `ParameterEstimates` is ready for the storage of the launch time, t, and the estimates $(\hat{M}, \hat{p}_c, \hat{q}_c, \hat{p}, \hat{q})$ (see Figure 3). Once completed the analysis for all the specialties, the two resulting matrices will be easily exported to csv or xls files.

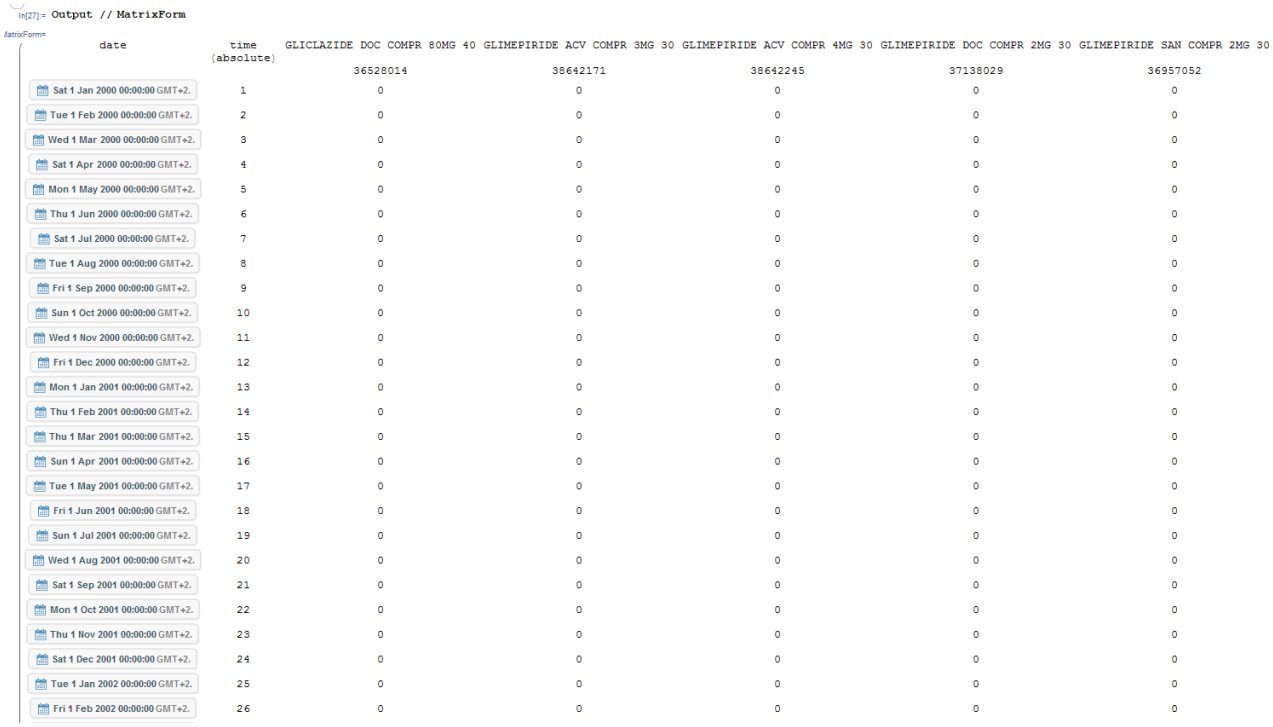
```
Export["C:\\PATH\\finalParameterEstimates.xls", ParameterEstimates];
Export["C:\\PATH\\finalForecasts.xls", Output];
```

3 Model fitting

3.1 Cumulative data

Model fitting requires a preliminary definition of the model structures (as explained in Appendix A). For specialties launched into the market before the data origin, a special approach based on density estimation is required and it will be described in the next Subsection. For most of the specialties, however, data cover also the beginning of the lifecycle. In that case, we can exploit the closed forms of the solutions for the different model structures and proceed with the estimation of parameters by fitting the models to cumulative data. These model structures for the BM, the GGM, the GGM with a rectangular shock, the GGM with an exponential shock, the GGM with both an exponential and a rectangular shock are defined as follows:

Figure 2: Structure for predicted values in matrix `Output` after the initialization commands (for space reasons, only the first 29 rows and 7 columns are displayed).



In[27]:= Output // MatrixForm

Out[27]//MatrixForm=

date	time (absolute)	GLICLAZIDE DOC COMPR 80MG 40	GLIMEPIRIDE ACV COMPR 3MG 30	GLIMEPIRIDE ACV COMPR 4MG 30	GLIMEPIRIDE DOC COMPR 2MG 30	GLIMEPIRIDE SAN COMPR 2MG 30
		36528014	38642171	38642245	37138029	36957052
Sat 1 Jan 2000 00:00:00 GMT+2.	1	0	0	0	0	0
Tue 1 Feb 2000 00:00:00 GMT+2.	2	0	0	0	0	0
Wed 1 Mar 2000 00:00:00 GMT+2.	3	0	0	0	0	0
Sat 1 Apr 2000 00:00:00 GMT+2.	4	0	0	0	0	0
Mon 1 May 2000 00:00:00 GMT+2.	5	0	0	0	0	0
Thu 1 Jun 2000 00:00:00 GMT+2.	6	0	0	0	0	0
Sat 1 Jul 2000 00:00:00 GMT+2.	7	0	0	0	0	0
Tue 1 Aug 2000 00:00:00 GMT+2.	8	0	0	0	0	0
Fri 1 Sep 2000 00:00:00 GMT+2.	9	0	0	0	0	0
Sun 1 Oct 2000 00:00:00 GMT+2.	10	0	0	0	0	0
Wed 1 Nov 2000 00:00:00 GMT+2.	11	0	0	0	0	0
Fri 1 Dec 2000 00:00:00 GMT+2.	12	0	0	0	0	0
Mon 1 Jan 2001 00:00:00 GMT+2.	13	0	0	0	0	0
Thu 1 Feb 2001 00:00:00 GMT+2.	14	0	0	0	0	0
Thu 1 Mar 2001 00:00:00 GMT+2.	15	0	0	0	0	0
Sun 1 Apr 2001 00:00:00 GMT+2.	16	0	0	0	0	0
Tue 1 May 2001 00:00:00 GMT+2.	17	0	0	0	0	0
Fri 1 Jun 2001 00:00:00 GMT+2.	18	0	0	0	0	0
Sun 1 Jul 2001 00:00:00 GMT+2.	19	0	0	0	0	0
Wed 1 Aug 2001 00:00:00 GMT+2.	20	0	0	0	0	0
Sat 1 Sep 2001 00:00:00 GMT+2.	21	0	0	0	0	0
Mon 1 Oct 2001 00:00:00 GMT+2.	22	0	0	0	0	0
Thu 1 Nov 2001 00:00:00 GMT+2.	23	0	0	0	0	0
Sat 1 Dec 2001 00:00:00 GMT+2.	24	0	0	0	0	0
Tue 1 Jan 2002 00:00:00 GMT+2.	25	0	0	0	0	0
Fri 1 Feb 2002 00:00:00 GMT+2.	26	0	0	0	0	0

Figure 3: Structure for predicted values in matrix `ParameterEstimates` after the initialization commands.

```
ParameterEstimates // MatrixForm
```

name	t_init	M	pc	qc	p	q
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

```
modBASS = M*((1 - Exp[-(p+q)*t])/(1 + (q/p)*Exp[-(p+q)*t]));
modGGM = M*Sqrt[(1 - Exp[-(pc+qc)*t])/(1 + (qc/pc)*Exp[-(pc+qc)*t])]*
    ((1 - Exp[-(p+q)*t])/(1 + (q/p)*Exp[-(p+q)*t]));
modGGMR1 = Piecewise[{{M* Sqrt[(1 - Exp[-(pc+qc)*t])/(1 + (qc/pc)*Exp[-(pc+qc)*t])]*
    ((1 - Exp[-(p+q)*(t + c1*(b1-a1))])/(1 + (q/p)* Exp[-(p+q)*(t + c1*(b1-a1))])),
    t > b1}, {M* Sqrt[(1 - Exp[-(pc+qc)*t])/(1 + (qc/pc)*Exp[-(pc+qc)*t])]*((1
    - Exp[-(p+q)*(t + c1*(t-a1))])/(1 + (q/p)* Exp[-(p+q)*(t + c1*(t-a1))])),
    a1 <= t <= b1}, {M* Sqrt[(1 - Exp[-(pc+qc)*t])/(1 + (qc/pc)*Exp[-(pc+qc)*t])]*
   ((1 - Exp[-(p+q)*t])/(1 + (q/p)*Exp[-(p+q)*t])), t < a1}}];
modGGME1 = Piecewise[{{M* Sqrt[(1 - Exp[-(pc+qc)*t])/(1 + (qc/pc)*Exp[-(pc+qc)*t])]*
   ((1 - Exp[-(p + q)*(t + (c1/b1)*(Exp[b1*(t-a1)] - 1))])/(1 + (q/p)*
    Exp[-(p+q)*(t + (c1/b1)*(Exp[b1*(t-a1)] - 1) )])), a1 <= t },
   {M* Sqrt[(1 - Exp[-(pc+qc)*t])/(1 + (qc/pc)*Exp[-(pc+qc)*t])]*
   ((1 - Exp[-(p+q)*t])/(1 + (q/p)*Exp[-(p+q)*t])), t < a1}}];
modGGME1R1 = Piecewise[{{M*Sqrt[(1 - Exp[-(pc+qc)*t])/(1 + (qc/pc)*Exp[-(pc+qc)*t])]*
   ((1 - Exp[-(p+q)*(t + c2*(b2-a2))])/(1 + (q/p)* Exp[-(p+q)*(t + c2*(b2-a2))])),
    t > b2}, {M* Sqrt[(1 - Exp[-(pc+qc)*t])/(1 + (qc/pc)*Exp[-(pc+qc)*t])]*
   ((1 - Exp[-(p+q)*(t + c2*(t-a2))])/(1 + (q/p)* Exp[-(p+q)*(t + c2*(t-a2))])),
    a2 <= t <= b2}, {M*Sqrt[(1 - Exp[-(pc+qc)*t])/(1 + (qc/pc)*Exp[-(pc+qc)*t])]*
   ((1 - Exp[-(p + q)*(t + (c1/b1)*(Exp[b1*(t-a1)] - 1))])/(1 + (q/p)*
    Exp[-(p+q)*(t + (c1/b1)*(Exp[b1*(t-a1)] - 1) )])), a1 <= t < a2},
    {M* Sqrt[(1 - Exp[-(pc+qc)*t])/(1 + (qc/pc)*Exp[-(pc+qc)*t])]*
   ((1 - Exp[-(p+q)*t])/(1 + (q/p)*Exp[-(p+q)*t])), t < a1}}];
```

The aim is to obtain estimated values for the main set of 5 parameters of the GGM (M, p_c, q_c, p, q), in order to use them to forecast the parameters of an emerging drug, given its expected launch time. We observe, however, that, for some time series, it may be difficult to provide good initial values to obtain directly a GGM fitting. For this reason, a preliminary estimation of a simpler BM, although unprecise, may give some indication about the size of the market potential, M, and the adoption evolution, (p, q). The estimated values for a BM may be subsequently used as initial values for three parameters of the GGM. This is the reason why we suggest to start always with a BM fitting.

Moreover, very often, we observe that the main lifecycle is perturbed by exogenous factors unrelated to the evolutionary path. By fitting a GGM, the effect of these local perturbations would affect the estimation of (M, p_c, q_c, p, q), thus introducing a noise for subsequent steps—when those estimates will be used—and also reducing the reliability of the forecasts. The aim of fitting a GGM with appropriate shocks is thus to isolate external shocks from the main trajectory of the lifecycle of the drug's sales (for example, an external factor may be due to an aggressive pressure from the pharmaceutical industry to physicians). Below an example will be shown to see how the trajectory obtained by fitting a GGM with shock may represent a better description of the data (with respect to a GGM without shock) and how the corresponding parameters estimates may be different.

In order to apply previous models to specialties' observations, for each series it is necessary to extract valid data (eliminating the null values for sales before the

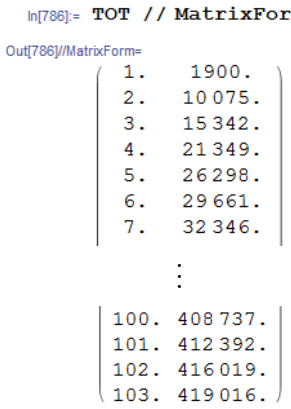
product's launch) and build cumulative observations. The counter `i` (ranging from 3 to the number of columns, `ncol`) allows exploring all the specialties with a common set of commands that are a function of its current value.

```
 i = 3;
serieTOT = data[[4 ;; nrow, i]];
d0 = Differences[Position[serieTOT, 0.‘]];
If[serieTOT[[1]] > 0, {initial = 1, Print["density?", data[[2, i]]]},
    If[Max[d0] == 1, {initial = Length[d0] + 2,
         Table[dataTEMP[[j, i]] = "", {j, 4, initial - 1 + 3}]},
    {initial = Position[d0, Max[d0]][[1]][[1]] + 1,
         Table[dataTEMP[[j, i]] = "", {j, 4, initial - 1 + 3}]} ]];
serieTOT = dataTEMP[[4 ;; nrow, i]];
len = nrow - 2 - initial;
TOT = Transpose[{data[[4 ;; 3 + len, 2]],
         Accumulate[serieTOT[[initial ;; initial + len - 1]]]}];
Nd = len;
label = ToString[data[[1, i]]]
  Out:  GLICLAZIDE DOC COMPR 80MG 40
```

The conditional command allows highlighting whether the data origin follows the launch time (in that case a density estimation is suggested). Otherwise, the launch time is calculated (and its value is saved as `initial`). The use of a provisional `dataTEMP` matrix, for assignments useful for data matrix building, allows to possibly repeat the analysis, thus executing again these commands for the same `i` value within the same session, without modifying the original data matrix. Cumulative values are saved in the matrix `TOT`, whose first column contains a relative timing, t, starting from 1 to `len` (the number of actual observations in the time series after removing initial zeros). In the end, a string with the name of the specialty, `label,` is built to be used in subsequent graphical representations. Figure 4 shows the matrix `TOT` built with previous commands.

The next commands allow effective estimation and require a specification of appropriate initial values to obtain convergence. The output produced contains parameter estimates and marginal confidence intervals, an ANOVA table, the R^2 and the squared correlation coefficient between observed instantaneous values and corresponding fitted values. Finally, a plot with observed instantaneous values and corresponding fitted values is produced.

Figure 4: Matrix `TOT` built with cumulative data for model estimation (i=3).




```
modfitBASS = NonlinearModelFit[TOT, modBASS,
  {{M, 100000}, {p, 0.01}, {q, 0.2}}, {t}, MaxIterations -> 10000]
modfitBASS["ParameterConfidenceIntervalTable"]
modfitBASS["ANOVATable"]
s2T = modfitBASS["ANOVATableSumsOfSquares"];
RsquacorrBASS = (s2T[[1]] - s2T[[3]] + s2T[[4]])/s2T[[4]]
g = modfitBASS["Function"];
FitValT = g[#] & /@ arg;
FitBASS = Transpose[{arg, Flatten[{FitValT[[1]], Differences[FitValT]}]}];
aT = Drop[FitBASS, -(Pr - Nd)][[All, 2]];
bT = serieTOT[[initial ;; initial + len - 1]];
RHO2BASST = (Correlation[bT, aT])^2
ListPlot[{bT, FitBASS}, PlotMarkers -> {[FilledSquare], Null},
  PlotStyle -> {{Blue, PointSize[Medium]}, {Black, Thick}},
  Joined -> {True, True}, Frame -> True,
  PlotRange -> {{0, Nd*1.2}, {0, Max[bT]*1.1}}, PlotLabel -> label,
  PlotLegend -> {"data", "BM"}, LegendPosition -> {0.4, 0.25},
  LegendSize -> 0.3, LegendBorderSpace -> 0.7, ShadowOffset -> 0,
  LegendTextSpace -> 2]
```

The range covered by the axes in the plot is automatically evaluated as a function of the actual number of observations in the specific time series, `Nd`, and of the maximum value of instantaneous sales (thus preventing a large portion of the picture to be wasted in the case of small `Nd` values with respect to `Pr`). The position of the legend may not be adequate if it overlaps a useful portion of the plot, but the coordinates on the `LegendPosition` command can be easily modified.[1]

Similar commands are written to fit the GGM (showing estimates and plotting the fitted trajectory, with a comparison with the BM). See, as an example, the corresponding output in Figure 5.

[1] Both coordinates of the `LegendPosition` command span the interval $[-1, +1]$.

For the examined specialty, the GGM provides a quite accurate description. However, other specialties may require further analysis through an intervention enrichment. For example, Figure 6 shows a situation where the GGM and the BM are almost overlapping and both are inadequate to describe the slow take-off in the sales of the specialty (the figure shows also the GGM parameter estimates). In that case, a GGM with a rectangular shock slowing the sales from the beginning until around t=60 has a much better performance and gives more reliable forecasts. Figure 7 presents the commands and the first part of the output, while in Figure 8 the plot of the fitted trajectory is shown.

Observe that the parameter estimates of (M, p_c, q_c, p, q) in Figure 7 are very different from the values in Figure 6, which as anticipated, are biased because the GGM is not an adequate structure. Since reliable parameter estimates are the main aim of this analysis—because they convey information on the features of the therapeutic category under study—the model choice represents a crucial step. With reference to this issue, usually it is quite difficult to decide in advance whether a rectangular shock would fit better than an exponential one. Or whether a "jump" in observations would be described better by a negative shock ending at that point or by a positive shock starting at the jump time. A good set of initial values aimed at exploring all the possibilities is, for this reason, a sensible approach.

Once an appropriate model has been identified, parameter estimates and forecasted values can be stored in matrices `ParameterEstimates` and Output, respectively. In the following command, the red expressions have to be set to declare the final model choice (either `FitGGM` or `FitGGMR` or `FitGGME`, etc.)

```
PREDTOT = FitGGMR;
Table[Output[[j, i]] = PREDTOT[[j - initial - 2, 2]], {j, initial + 3, Pr + 3}];
ParameterEstimates[[i, 1 ]] = label;
ParameterEstimates[[i, 2 ]] = initial;
Table[ParameterEstimates[[i, h]] =
  modfitGGMR["ParameterConfidenceIntervalTableEntries"][[h - 2, 1]], {h, 3, 7}];
```

3.2 Instantaneous data

For specialties launched into the market before the data origin, we cannot use cumulative data as a response function in the regression model. An alternative model specification uses directly instantaneous sales, while the mean trajectory function, $z(\beta, t)$ has to be accordingly replaced by $z(\beta, t + 0.5) - z(\beta, t - 0.5)$ (further details about the estimation procedure with instantaneous data could be found, e.g., in Guseo and Mortarino, 2015, Supplementary Material).

The definition of the model structures is here presented for the BM and the GGM without intervention functions. Extensions may be obtained by adding rectangular or exponential shocks.

Figure 5: Output for the GGM estimation (i=3). It shows: i) the fitted model (in Mathematica form); ii) parameter estimates (with standard errors and marginal confidence intervals); iii) the ANOVA table; iv) the R^2 of the model (with cumulative observations as a response); v) the squared correlation coefficient between observed instantaneous values and corresponding fitted values; vi) the plot of the instantaneous data and the corresponding fitted trajectory (both for the GGM and the simpler BM).

i)

$$\texttt{FittedModel}\left[\frac{530595.\left(1-e^{-134.142\,t}\right)\sqrt{\frac{1-e^{-0.0447216\,t}}{1+57.5745\,e^{-\ll 20\gg t}}}}{1+320.384\,e^{-134.142\,t}}\right]$$

ii)

	Estimate	Standard Error	Confidence Interval
M	530595.	8268.04	{514187., 547003.}
pc	0.000763499	0.0000159874	{0.000731772, 0.000795225}
qc	0.0439581	0.000701365	{0.0425662, 0.0453499}
p	0.417387	5.2287×10^{-64}	{0.417387, 0.417387}
q	133.724	1.53161×10^{-64}	{133.724, 133.724}

iii)

	DF	SS	MS
Model	5	5.92765×10^{12}	1.18553×10^{12}
Error	98	1.64099×10^{9}	1.67448×10^{7}
Uncorrected Total	103	5.92929×10^{12}	
Corrected Total	102	1.50103×10^{12}	

iv) `0.998907`

v) `0.374111`

vi)

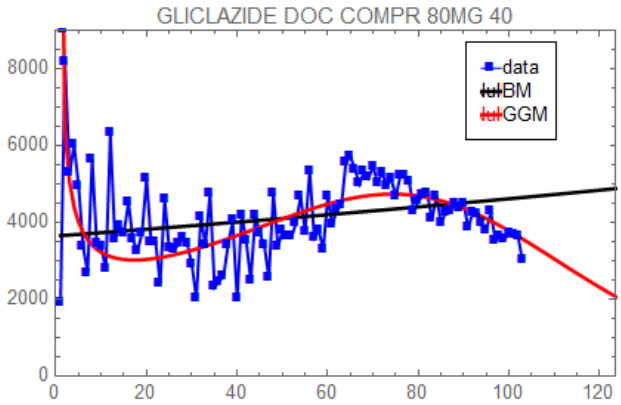
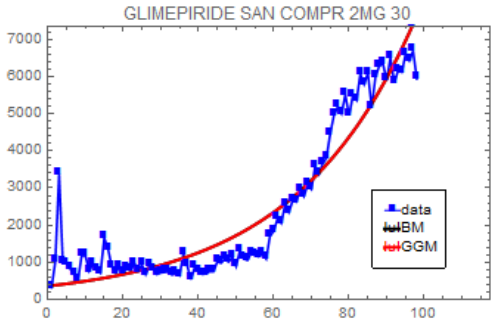


Figure 6: An inadequate performance for the GGM (i=7).



	Estimate	Standard Error	Confidence Interval
M	5.40638×10^9	6.07975×10^{-8}	$\{5.40638 \times 10^9, 5.40638 \times 10^9\}$
pc	12.5804	1097.99	{−2167.82, 2192.98}
qc	−14.6448	1194.97	{−2387.62, 2358.33}
p	7.41974×10^{-8}	6.26488×10^{-6}	{−0.0000123666, 0.000012515}
q	0.0310015	0.000693754	{0.0296238, 0.0323791}

```
modBASSDENS = M*((1 - Exp[-(p+q)*(t+0.5)])/(1 + (q/p)*Exp[-(p+q)*(t+0.5)]))
   - M*((1 - Exp[-(p+q)*(t-0.5)])/(1 + (q/p)*Exp[-(p+q)*(t-0.5)]));
modGGMDENS =
   (M*((1 - Exp[-(pc+qc)*(t+0.5)])/(1 + (qc/pc)* Exp[-(pc+qc)*(t+0.5)]))^(1/2)
   *(1 - Exp[-(p+q)*(t+0.5)])/(1 + (q/p)* Exp[-(p+q)*(t+0.5)]))
   - (M*((1 - Exp[-(pc+qc)*(t-0.5)])/(1 + (qc/pc)* Exp[-(pc+qc)*(t-0.5)]))^(1/2)
   *(1 - Exp[-(p+q)*(t-0.5)])/(1 + (q/p)*Exp[-(p+q)*(t-0.5)]));
```

The commands to build the data matrix for the regression models are different since we use instantaneous data. In any case, we can start with the same block of commands described at p. 8. If we examine here column 8, we obtain:

```
i = 8;
serieTOT = data[[4 ;; nrow, i]];
⋮
label = ToString[data[[1, i]]]
   Out:  density?  1 March 1996
   Out:  GLUCOMIDE CAPS 500MG 40
```

A suggestion to move to density is printed, together with the information about the launch time stored in the second line of the dataset. Given this information, the matrix for the regression model can be built with instantaneous data, while the corresponding relative timing, has to be changed putting its start at the indicated launch time.

```
Nd = nrow - 3;
serieTOT = data[[4 ;; nrow, i]];
TOT = Transpose[{data[[4 ;; nrow, 2]], serieTOT}];
decrem = DateDifference[data[[2, i]], data[[4, 1]], "Month"][[1]];
Table[TOT[[j, 1]] = TOT[[j, 1]] + decrem, j, 1, Nd];
Prtemp = Pr + decrem;
argtemp = Table[i, {i, 1, Prtemp, 1}];
```

Figure 9 shows the matrix `TOT` with instantaneous data built with previous commands. The time shift, `decrem`, between the launch time and the data origin is

Figure 7: Input —and output for the fitting of a GGM with an initial negative rectangular shock (i=7). After i) the fitted model (in Mathematica form), ii) parameter estimates and iii) the ANOVA table, iv) the R^2 of the model and v) the squared correlation coefficient between observed instantaneous values and corresponding fitted values are displayed.

```
modfitGGMR = NonlinearModelFit[TOT, modGGMR1,
  {{M, 705 000}, {pc, 0.15}, {qc, 0.2}, {p, 0.0015}, {q, 0.04},
   {c1, -0.2}, {a1, 10}, {b1, 60}}, {t}, MaxIterations → 30 000]
```

i)

```
FittedModel[ { «1» } ]
```

```
modfitGGMR["ParameterConfidenceIntervalTable"]
modfitGGMR["ANOVATable"]
s2TR = modfitGGMR["ANOVATableSumsOfSquares"];
RsquacorrGGMR = (s2TR[[1]] - s2TR[[3]] + s2TR[[4]]) / s2TR[[4]]
gTR = modfitGGMR["Function"];
FitValRT = gTR[#] & /@ arg;
FitGGMR = {arg, Flatten[{FitValRT[[1]], Differences[FitValRT]}]};
FitGGMR = Transpose[FitGGMR];
aRT = Drop[FitGGMR, -(Pr - Nd)][[All, 2]];
RHO2GGMR = (Correlation[bT, aRT])^2
```

ii)

	Estimate	Standard Error	Confidence Interval
M	684 821.	56 793.9	{571 990., 797 652.}
pc	0.000752184	0.0290395	{−0.0569398, 0.0584442}
qc	4.73636	22.0831	{−39.1355, 48.6082}
p	0.00124195	0.000097827	{0.0010476, 0.0014363}
q	0.0414937	0.00124875	{0.0390129, 0.0439746}
c1	−0.585401	0.00792999	{−0.601155, −0.569647}
a1	14.829	0.779933	{13.2796, 16.3785}
b1	63.4244	0.334711	{62.7594, 64.0894}

iii)

	DF	SS	MS
Model	8	8.70342×10^{11}	1.08793×10^{11}
Error	90	1.32007×10^{8}	1.46675×10^{6}
Uncorrected Total	98	8.70474×10^{11}	
Corrected Total	97	3.88422×10^{11}	

iv) `0.99966`

v) `0.953358`

calculated automatically through the function `DateDifference` giving the gap in number of months. An extension of the the number of future predicted values—in order to reach also in this case the ending point, here February 2019—is obtained by `Prtemp` and `argtemp` (these are extended versions of `Pr` and `arg`, respectively, adding `decrem` extra observations to compensate the initial time shift).

The commands for model fitting, output presentation and graphical comparison is similar to the case seen for cumulative data, but some some changes are required to deal with the time shift. In Figures 10 and 11, the modified commands and the

Figure 8: Plot of the fitted trajectory of a GGM with a rectangular shock (i=7).

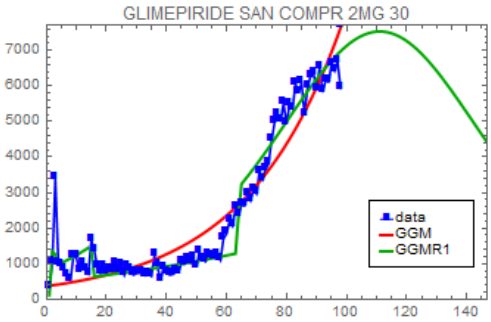


Figure 9: Matrix `TOT` built with instantaneous data for model estimation through density (i=8).

```
In[804]:= TOT // MatrixForm
```

$$\begin{pmatrix} 47. & 38\,610. \\ 48. & 47\,489. \\ 49. & 55\,401. \\ 50. & 45\,658. \\ 51. & 56\,658. \\ 52. & 53\,152. \\ 53. & 53\,474. \\ 54. & 48\,133. \\ 55. & 55\,490. \\ 56. & 56\,205. \\ 57. & 53\,244. \end{pmatrix}$$

$$\vdots$$

$$\begin{pmatrix} 180. & 252. \\ 181. & 55. \\ 182. & 748. \\ 183. & 538. \\ 184. & 384. \\ 185. & 68. \\ 186. & 15. \\ 187. & 22. \\ 188. & 18. \\ 189. & 4. \\ 190. & 1. \end{pmatrix}$$

$$\vdots$$

$$\begin{pmatrix} 218. & 0. \\ 219. & 0. \\ 220. & 0. \\ 221. & 0. \\ 222. & 0. \end{pmatrix}$$

Figure 10: Input and output for the fitting of a GGM through density estimation (i=8). After i) the fitted model (in Mathematica form), ii) parameter estimates and iii) the ANOVA table, iv) the R^2 of the model and v) the squared correlation coefficient between observed instantaneous values and corresponding fitted values are displayed.

```
modfitGGM = NonlinearModelFit[TOT, modGGMDENS,
  {{M, 1000000}, {pc, 0.005}, {qc, -0.05}, {p, 0.0028}, {q, 0.14}}, {t}, MaxIterations → 30000]
```

i)

$$\text{FittedModel}\left[-\frac{5.78444\times10^6\sqrt{\frac{1-e^{-\ll19\gg(\ll1\gg)}}{1+20.8993\,e^{\ll1\gg}}}\left(1-e^{-0.0272716(-0.5+t)}\right)}{1+3.05557\,e^{-0.0272716(-0.5+t)}}+\frac{5.78444\times10^6\sqrt{\frac{1-\ll1\gg}{1+\ll1\gg}}\left(1-e^{-\ll21\gg(\ll1\gg)}\right)}{1+3.05557\,e^{-\ll21\gg(\ll1\gg)}}\right]$$

```
modfitGGM["ParameterConfidenceIntervalTable"]
modfitGGM["ANOVATable"]
s2T = modfitGGM["ANOVATableSumsOfSquares"];
RsquacorrGGM = (s2T[[1]] - s2T[[3]] + s2T[[4]]) / s2T[[4]]
g = modfitGGM["Function"];
FitValT = g[#] & /@ argtemp;
FitValT = Drop[FitValT, decrem];
FitGGM = Transpose[{Drop[argtemp, decrem], FitValT}];
aT = Take[FitGGM, Nd - 1][[All, 2]];
bT = Take[serieTOT, Nd - 1];
RHO2GGM = (Correlation[bT, aT]) ^ 2
```

ii)

	Estimate	Standard Error	Confidence Interval
M	5.78444×10^6	146386.	$\{5.49548\times10^6, 6.07339\times10^6\}$
pc	0.782657	0.00318955	{0.776361, 0.788953}
qc	16.357	0.00317885	{16.3507, 16.3633}
p	0.00672447	0.000494802	{0.00574777, 0.00770118}
q	0.0205471	0.0014544	{0.0176762, 0.023418}

iii)

	DF	SS	MS
Model	5	1.18788×10^{11}	2.37576×10^{10}
Error	171	1.83144×10^9	1.07102×10^7
Uncorrected Total	176	1.20619×10^{11}	
Corrected Total	175	5.35066×10^{10}	

iv) `0.965772`

v) `0.966412`

corresponding output are shown.

Parameter estimates and forecasted values' storage can be obtained with commands identical to those presented at the end of previous Subsection for cumulative data regression.

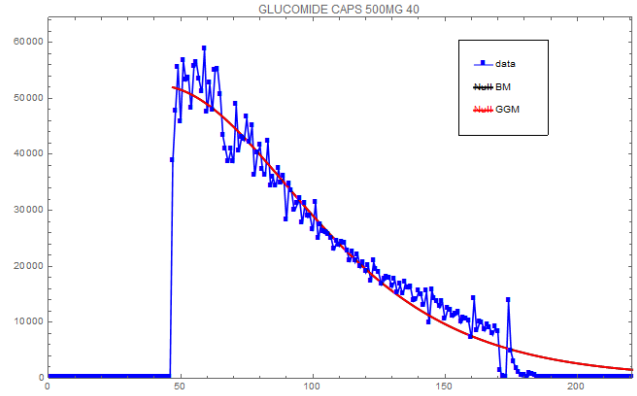
4 A not-so-rare estimation issue

The `NonlinearModelFit` command performs an unconstrained minimization of the residual sum of squares. Given the search algorithms, it may sometimes happen that parameter estimates are outside the expected range.

A relevant issue arises when this happens for the q_c parameter (which, with p_c, determines the growth speed of the market potential towards its limit, M, see Equation (8)). We are interested in the value obtained when t goes to $+\infty$ (the ultimate value of cumulative sales at the end of the life cycle). In the case $(p_c+q_c) > 0$, M actually represents the limit of $m(t)$ as t goes to infinity. Conversely, for $p_c > 0$

Figure 11: Input commands and plot of the fitted trajectory of a GGM through density estimation (i=8).

```
ListPlot[{Flatten[Prepend[serieTOT, Table[0, {j, 1, decrem}]]], FitBASS, FitGGM}, Joined → {True, True, True},
 PlotMarkers → {■, Null, Null}, PlotStyle → {{Blue, PointSize[Medium]}, {Black, Thick}, {Red, Thick}},
 Frame → True, PlotRange → {{0, Prtemp * 0.8}, {0, Max[bT] * 1.1}}, PlotLabel → label, PlotLabel → label,
 PlotLegend → {"data", "BM", "GGM"}, LegendPosition → {0.4, 0.20}, LegendSize → 0.3, LegendBorderSpace → 0.7,
 ShadowOffset → 0, LegendTextSpace → 2]
```



and $(p_c + q_c) < 0$,

$$\lim_{t\to\infty} M\sqrt{\frac{1-e^{-(p_c+q_c)t}}{1+\frac{q_c}{p_c}e^{-(p_c+q_c)t}}} = M\sqrt{-\frac{p_c}{q_c}}.$$

This consideration affects the interpretation and the meaning of parameter M for a negative q_c estimate (the situation of a negative p_c usually is less relevant, since its magnitude often is much smaller than q_c and it is quite rare that a negative p_c, entails $(p_c + q_c) < 0$).

Let us illustrate this aspect with an example obtained for `i`=7. If for the GGM model with a rectangular shock (as in Figure 7), we start from a different set of initial values, we obtain the estimates reported in Figure 12. The global performance of the model is similar for the two alternative solutions but we observe a huge difference in the estimation for M, whose interpretation for the solution of Figure 12 might suggest we are dealing with a very high-performing drug.

Given a negative q_c estimate, we suggest then plotting the estimated trajectory of the market potential function (see Figure 13). This plot suggests a more reliable value for the value of $\hat{M}$, by taking the asymptotic value (that is, the limit for $t \to +\infty$) of the plotted function[2]. Moreover, as shown in Figure 14, we can identify corresponding p_c and q_c values to mimic the trajectory of the former function (Figure 13, blue trajectory) with the new M value. This simple graphical method provides better initial values to be used in the `NonlinearModelFit` command to reach a good convergence point.

[2]Clicking the right mouse button on the plot displays a context menu from which we can choose `Get Coordinates` in order see approximate coordinate values of the mouse position.

Figure 12: Output for the fitting of a GGM with an initial negative rectangular shock with a different set of initial values (i=7). It shows parameter estimates, the ANOVA table, the R^2 of the model, and the squared correlation coefficient between observed instantaneous values and corresponding fitted values.

	Estimate	Standard Error	Confidence Interval
M	2.36142×10^7	8.33613×10^{-10}	$\{2.36142 \times 10^7, 2.36142 \times 10^7\}$
pc	0.00148888	0.0103833	{−0.0191394, 0.0221172}
qc	−1.77256	12.3653	{−26.3383, 22.7932}
p	0.00124157	0.0000978253	{0.00104723, 0.00143592}
q	0.0415083	0.00125031	{0.0390243, 0.0439922}
c1	−0.58534	0.00794253	{−0.601119, −0.56956}
a1	14.846	0.782405	{13.2916, 16.4004}
b1	63.4253	0.335164	{62.7594, 64.0911}

	DF	SS	MS
Model	8	8.70341×10^{11}	1.08793×10^{11}
Error	90	1.32332×10^8	1.47035×10^6
Uncorrected Total	98	8.70474×10^{11}	
Corrected Total	97	3.88422×10^{11}	

```
0.999659
```

```
0.950103
```

Figure 13: Input and output for market potential function plotting in the case of $\hat{q}_c < 0$ (i=7), in order to find the limit for $t \to +\infty$.

```
marPot[t_, M_, pc_, qc_] = M * Sqrt[(1 - Exp[-(pc + qc) * t]) / (1 + (qc / pc) * Exp[-(pc + qc) * t])];

Plot[{marPot[t, 2.3614151586832248`*^7, 0.001488878361059044`, -1.7725609479451339`]},
 {t, -100, 100}, PlotStyle → {{Red, Thick}}, PlotRange → {{-100, 100}, {0, 30 000 000}}]
```

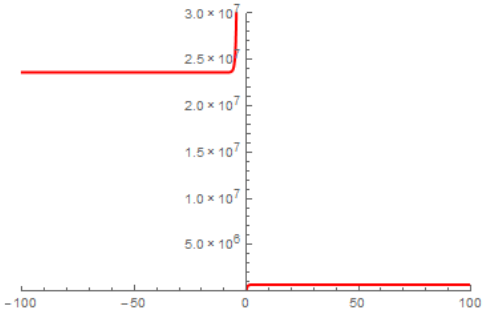


Figure 14: Input and output for market potential function plotting (red trajectory) and identification of alternative values for p_c and q_c to mimic the estimated function with an appropriate approximation (blue trajectory).

```
Plot[{marPot[t, 2.3614151586832248`*^7, 0.001488878361059044`, -1.7725609479451339`],
  marPot[t, 681500, 0.95, 0.8]}, {t, -100, 100}, PlotStyle → {{Red, Thick}, Blue},
 PlotRange → {{0, 10}, {0, 1000000}}]
```

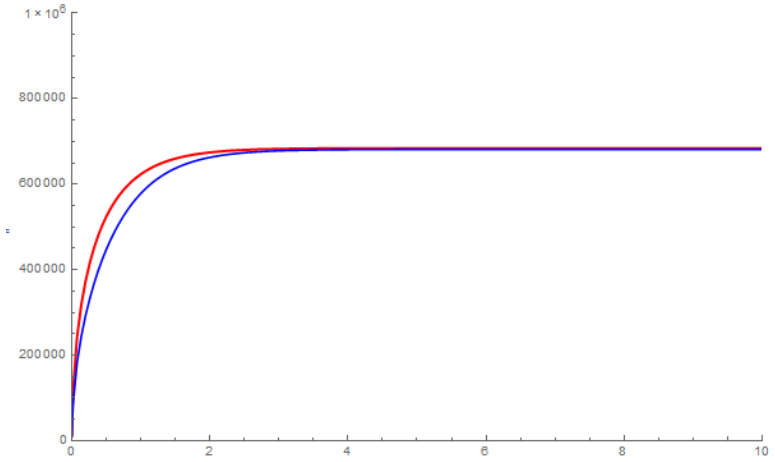


Figure 15: Final `ParameterEstimates` matrix.

```
ParameterEstimates // MatrixForm
```

name	t_init	M	pc	qc	p	q
GLICLAZIDE DOC COMPR 80MG 40	74	530595.	0.000763499	0.0439581	0.417387	133.724
GLIMEPIRIDE ACV COMPR 3MG 30	114	313874.	0.000823627	0.112594	0.00346888	0.0241743
GLIMEPIRIDE ACV COMPR 4MG 30	114	310337.	0.000982217	0.115949	0.00305405	0.0311228
GLIMEPIRIDE DOC COMPR 2MG 30	93	29682.3	247.324	587.751	0.00252427	0.159492
GLIMEPIRIDE SAN COMPR 2MG 30	79	684821.	0.000752184	4.73636	0.00124195	0.0414937
GLUCOMIDE CAPS 500MG 40	1	5.78444×10^6	0.782657	16.357	0.00672447	0.0205471

5 Final remarks

The final matrix with all time series parameter estimates is shown, for the simple dataset here analyzed, in Figure 15. This matrix, together with `Output` (containing the predicted values for all the specialties, Figure 16), represents the aim of this phase of the analysis. Each drug is summarized by its absolute launch time and by the set of five parameters describing its lifecycle.

As described in Guseo et al. (2016), each parameter estimate will be regressed on the initial time t to produce adequate models describing the dynamic features of the category under study. For an application to Italian data (Jommi et al., 2016), about 200 series have been analyzed. A preliminary inspection of corresponding German data, for example, show that we should be able to deal with almost 2000 time series (even if some of them may be too short to obtain reliable estimates). It is clear that the computational effort is, in that case, quite relevant.

Figure 16: Final `Ouput` matrix.

Output // MatrixForm

date	time (absolute)	GLICLAZIDE 80MG 40	DOC COMPR 80MG 40	GLIMEPIRIDE ACV COMPR 3MG 30	GLIMEPIRIDE ACV COMPR 4MG 30	GLIMEPIRIDE DOC COMPR 2MG 30	GLIMEPIRIDE SAN COMPR 2MG 30	GLUCOMIDE CAPS 500MG 40
		36528014		38642171	38642245	37138029	36957052	1 March 1996 26130029
Sat 1 Jan 2000 00:00:00 GMT+2.	1		0	0	0	0	0	51989.9
Tue 1 Feb 2000 00:00:00 GMT+2.	2		0	0	0	0	0	51863.9
Wed 1 Mar 2000 00:00:00 GMT+2.	3		0	0	0	0	0	51719.2
Sat 1 Apr 2000 00:00:00 GMT+2.	4		0	0	0	0	0	51555.9
Mon 1 May 2000 00:00:00 GMT+2.	5		0	0	0	0	0	51374.3
Thu 1 Jun 2000 00:00:00 GMT+2.	6		0	0	0	0	0	51174.6
Sat 1 Jul 2000 00:00:00 GMT+2.	7		0	0	0	0	0	50957.2
Tue 1 Aug 2000 00:00:00 GMT+2.	8		0	0	0	0	0	50722.4
Fri 1 Sep 2000 00:00:00 GMT+2.	9		0	0	0	0	0	50470.5
Sun 1 Oct 2000 00:00:00 GMT+2.	10		0	0	0	0	0	50201.8
Wed 1 Nov 2000 00:00:00 GMT+2.	11		0	0	0	0	0	49916.7
...								
Fri 1 Dec 2017 00:00:00 GMT+2.	216		1031.79	2335.83	2514.54	0.00280289	5501.11	502.164
Mon 1 Jan 2018 00:00:00 GMT+2.	217		992.446	2318.35	2481.4	0.00238366	5378.54	488.717
Thu 1 Feb 2018 00:00:00 GMT+2.	218		954.382	2300.26	2447.55	0.00202713	5255.27	475.629
Thu 1 Mar 2018 00:00:00 GMT+2.	219		917.571	2281.58	2413.03	0.00172392	5131.55	462.89
Sun 1 Apr 2018 00:00:00 GMT+2.	220		881.988	2262.33	2377.89	0.00146607	5007.63	450.49
Tue 1 May 2018 00:00:00 GMT+2.	221		847.607	2242.53	2342.18	0.00124679	4883.74	438.422
Fri 1 Jun 2018 00:00:00 GMT+2.	222		814.403	2222.18	2305.97	0.0010603	4760.1	426.675
Sun 1 Jul 2018 00:00:00 GMT+2.	223		782.348	2201.32	2269.31	0.000901711	4636.91	415.242
Wed 1 Aug 2018 00:00:00 GMT+2.	224		751.415	2179.97	2232.24	0.00076684	4514.38	404.114
Sat 1 Sep 2018 00:00:00 GMT+2.	225		721.575	2158.13	2194.82	0.000652142	4392.67	393.283
Mon 1 Oct 2018 00:00:00 GMT+2.	226		692.8	2135.85	2157.11	0.0005546	4271.98	382.741
Thu 1 Nov 2018 00:00:00 GMT+2.	227		665.063	2113.13	2119.15	0.000471647	4152.45	372.481
Sat 1 Dec 2018 00:00:00 GMT+2.	228		638.334	2090.01	2080.99	0.000401101	4034.23	362.495
Tue 1 Jan 2019 00:00:00 GMT+2.	229		612.586	2066.5	2042.69	0.000341108	3917.46	352.775
Fri 1 Feb 2019 00:00:00 GMT+2.	230		587.79	2042.64	2004.29	0.000290087	3802.27	343.316

Appendix A. Innovation diffusion models and statistical inference

The closed-form solution of the Bass Model (BM), proposed by Bass (1969), is given by

$$z(t) = m\,\frac{1-e^{-(p+q)t}}{1+\frac{q}{p}\,e^{-(p+q)t}}, \quad z(0)=0, \;\; t\geq 0, \;\; p,q>0, \tag{1}$$

and $z(t)=0$ for $t<0$, where $z(t)$ denotes the observed cumulative sales up to time t. The constraint $z(0)=0$ is the initial condition and m is the asymptotic constant market potential. The parameters p and q describe the innovative and the imitative effects, respectively.

The Generalized Bass Model (GBM), Bass et al. (1994), represents an important extension of the BM, introducing a general time-dependent integrable intervention function, $x(t)$, that is able to take into account possible effects of exogenous variables on the diffusion process, for example marketing mix variables. Its Equation is:

$$z'(t) = \left[p + q\frac{z(t)}{m}\right][m-z(t)]x(t), \tag{2}$$

and its closed-form solution is, under the initial condition $z(0)=0$,

$$z(t) = m\,\frac{1-e^{-(p+q)\int_0^t x(\tau)d\tau}}{1+\frac{q}{p}e^{-(p+q)\int_0^t x(\tau)d\tau}}, \quad t>0, \;\; p,q>0. \tag{3}$$

Notice that the GBM reduces to the Bass model, when $x(t)=1$, i.e. when there are no external interventions. The important effect of $x(t)$ is to anticipate ($x(t)>1$) or delay ($x(t)<1$) adoptions, but not to increase or decrease them. In other words, this function may represent all those strategies applied to control the timing of a diffusion process, but not its size.

In Guseo and Dalla Valle (2005) and Guseo et al. (2007) some parametric functions for the intervention function, $x(t)$, are suggested. A local perturbation, whose effect is strong at beginning and time decaying, may be modelled through an exponential function like

$$x(t) = 1 + ce^{b(t-a)}I_{t\geq a}, \tag{4}$$

where parameter c represents the depth and sign of intervention, b describes the persistence of the induced effect and is negative if the memory of this intervention is decaying to the stationary position (*mean reverting*) and a denotes the starting times of intervention, so that $(t-a)$ must be positive. A more stable perturbation acting on diffusion for a relatively long period, like institutional measures and policies, may be described by a rectangular function giving rise to the intervention function

$$x(t) = 1 + cI_{t\geq a}I_{t\leq b}. \tag{5}$$

Parameter c describes here the perturbation intensity and may be either positive or negative, while parameters a and b define the temporal interval in which the shock

occurs. Of course, the actual function $x(t)$ could be designed as a combination of one or many shocks as modelled in (4) and (5).

The Guseo Guidolin Model (GGM), Guseo and Guidolin (2009), extends both the BM and the GBM. The relevant new feature of the GGM is the general shape of latent market potential, $m(t)$, in contrast to the constant assumption m in the BM. Its general aggregate differential form is:

$$z'(t) = m(t)\left\{\left[p + q\frac{z(t)}{m(t)}\right]\left[1 - \frac{z(t)}{m(t)}\right]\right\}x_a(t) + z(t)\frac{m'(t)}{m(t)},\ z(0) = 0,\ t \geq 0, \quad (6)$$

with the usual constraint, $z(t) = 0$ for $t < 0$, where $z(t)$ denotes the cumulative sales, $m(t)$ the variable market potential, $x_a(t)$ an exogenous intervention function acting on the adoption process, and $z(t)m'(t)/m(t)$ a collective self–reinforcing effect that emphasizes or depresses sales on the basis of the sign of $m'(t)$. Parameters p and q denote the local dynamics of the adoption process.

The general closed-form solution to Equation (6) with an initial condition $z(0) = 0$ and $z(t) = 0$ for $t < 0$ is:

$$z(t) = m(t)\frac{1 - e^{-(p+q)\int_0^t x_a(\tau)d\tau}}{1 + \frac{q}{p}e^{-(p+q)\int_0^t x_a(\tau)d\tau}},\ t \geq 0,\ p, q > 0, \quad (7)$$

and zero elsewhere. Solution (7) does not depend upon special choices of $m(t)$ and $x(t)$. In Guseo and Guidolin (2009) special emphasis is given to the structure

$$m(t) = M\sqrt{\frac{1 - e^{-(p_c+q_c)t}}{1 + \frac{q_c}{p_c}e^{-(p_c+q_c)t}}}, \quad (8)$$

where p_c and q_c denote the communication parameters generating the nonconstant market potential, and M is the asymptotic market potential. In particular, in Guseo and Guidolin (2011) some explicit examples are proposed in pharmaceutical drug diffusions exhibiting saddles or slowdown effects relevant for marketers. These effects appear as significant drops after a period of rapid growth, followed by a recovery to the former peak. Notice that also the diffusion process under square root in (8) could be perturbed by a proper intervention function, $x_c(t)$, acting on the communication process.

An inferential methodology to estimate and test the performance of the models described in previous sub-sections may be implemented through a nonlinear regressive model, i.e.,

$$w(t) = z(\beta, t) + \varepsilon(t), \quad (9)$$

where $w(t)$ represents the observed cumulative sales data and $z(\beta, t)$ denotes the appropriate systematic mean rescaled cumulative distribution (either (1), or (3), or (7)), which is a function of time t and a vector of parameters β typical of the BM, GBM, or the GGM. The residual term $\varepsilon(t)$ is usually a white noise or a more complex stationary process, if seasonality and/or autoregressive aspects are included as stochastic components.

For estimation purposes, we apply a robust nonlinear least squares algorithm (NLS), which ignores the stochastic structure of $\varepsilon(t)$, under the Levenberg-Marquardt

correction of the Gauss-Newton recursive procedure; see, for instance, Seber and Wild (1989). If the residuals of the first stage do not follow a standard white noise pattern, in order to improve short-term prediction (when data are available), a SARMAX refinement could be performed.

Appendix B. Hints about Mathematica's commands sintax

Mathematica is a very powerful software. On the web there are many tutorials illustrating its several features (writing "Mathematica introduction" in Google we obtain about 1 million results). This Appendix is only meant to help readers not familiar with this software understanding the commands described in this report for this specific application.

The Mathematica prompt waits for user's commands, written in *cells*. In order to evaluate a cell's content, we need to press simultaneously SHIFT and ENTER (or to select a cell through its right bar, click the right mouse button and select the option "Evaluate Cell", which is particularly useful to evaluate multiple selected cells).

A helpful shortcut is obtained with CTRL + L, which in any point of the notebook creates a copy of the command written just above, useful to modify a command obtaining a similar one.

The names of all built-in Mathematica functions begin with capital letters. They tend not to abbreviate names, beyond what is normal mathematical practice, and multi-word names have internal capitalization. Some examples are: `Sin, Log, Sqrt, Random, Eigenvalues, NonlinearModelFit.`

Many operations can be performed through lists. A list is defined through curly brackets:

$$\{1, 2, 3, 4\}.$$

A matrix can be defined as a list of lists:

$$m = \{\{1, 2\}, \{3, 4\}\} \quad (\text{a } 2 \times 2 \text{ matrix}).$$

List indices start at 1, not 0. To access an element of a list, double square brackets are needed:

$$m[[2]][[2]] \quad (\text{is equivalent to } m[[2, 2]]).$$

If a subset of rows or columns is needed, we can specify a range as follows:

$$m[[5\ ;;10,\ 4]]),$$

or a subset with an internal list:

$$m[[5, \{6, 8, 9\}]].$$

Lists can also be created as follows

$$\text{Table}[x\hat{}2, \{x, 4, 12\}]$$

This last example has as its second argument, an "iterator list", which defines the iterator variable ("x"), and its range; you can specify either a maximum; or a min

and max; or min, max, and step size (whose default value is 1). These three iterator lists are equivalent: $\{x, 12\}$, $\{x, 1, 12\}$, $\{x, 1, 12, 1\}$. In general, the command `Table` is useful to repeat an operation a certain number of times (see, e.g., the initialization of matrices `Output` and `ParameterEstimates` described at p. 5).

The commands `Differences` and `Accumulate` can be applied to lists (vectors) to obtain successive differences of the elements in the list and successive accumulated totals of elements in the list, respectively. Another useful command is `Position`$[list, condition]$, which gives a list of the positions at which objects matching *condition* appear in *list*. For example,

$$\texttt{Position}[M, 5]$$

would produce a list of elements of matrix M whose values equal 5 (if none of the elements matches the *condition*, an empty list is produced).

An expression can be defined as:

$$\texttt{expr} = 3x + 7$$

Notice that if a command ends with a semicolon, it will not produce an output. This is particularly useful for assignments.

For the arguments of any function (either built-in or defined by the user), Mathematica uses single square brackets:

$$f[x, y].$$

Here is a function representing the same expression written for `expr`:

$$f[x\_] := 3x + 7$$

Note the "$x\_$", (called "x-blank") that represents any expression, and names it x; and the delayed-evaluation operator, ":=", which leaves the right-hand side in its symbolic form, instead of evaluating it immediately (which might cause a value to be substituted for x). We can now evaluate this function in various ways: $f[5]$, $f[t]$, $f[a + b]$ or represent it graphically in a specified domain:

$$\texttt{Plot}[f[x], \{x, -5, 5\}].$$

For functions with multiple definitions in different subdomains, it is easy to use the command Piecewise taking a list as its argument:

$$g[x\_] := \texttt{Piecewise}[\{\{x^2, x < 0\}, \{x, x > 0\}\}]$$

References

Bass, F. M. (1969). "A new product growth model for consumer durables". *Management Science*, 15, pp. 215–227.

Bass, F. M., Krishnan, T., and Jain, D. (1994). "Why the Bass model fits without decision variables". *Marketing Science*, 13, pp. 203–223.

Goodwin, P., Dyussekeneva, K., and Meeran, S. (2013). "The use of analogies in forecasting the annual sales of new electronics products". *IMA Journal of Management Mathematics*, 24, 4, pp. 407–422.

Goodwin, P., Meeran, S., and Dyussekeneva, K. (2014). "The challenges of pre-launch forecasting of adoption time series for new durable products.". *International Journal of Forecasting*, 30, 4, pp. 1082–1097.

Guseo, R. and Dalla Valle, A. (2005). "Oil and gas depletion: Diffusion models and forecasting under strategic intervention". *Statistical Methods and Applications*, 14, 3, pp. 375–387.

Guseo, R., Dalla Valle, A., Furlan, C., Guidolin, M., and Mortarino, C. (2016). "Pre-launch forecasting of a pharmaceutical drug". Manuscript submitted for publication.

Guseo, R., Dalla Valle, A., and Guidolin, M. (2007). "World oil depletion models: price effects compared with strategic or technological interventions". *Technological Forecasting and Social Change*, 74, 4, pp. 452–469.

Guseo, R. and Guidolin, M. (2009). "Modelling a dynamic market potential: A class of automata networks for diffusion of innovations". *Technological Forecasting and Social Change*, 76, 6, pp. 806–820.

Guseo, R. and Guidolin, M. (2011). "Market potential dynamics in innovation diffusion: modelling the synergy between two driving forces". *Technological Forecasting and Social Change*, 78, 1, pp. 13–24.

Guseo, R. and Mortarino, C. (2015). "Modeling competition between two pharmaceutical drugs using innovation diffusion models". *Annals of Applied Statistics*, 9, 4, pp. 2073–2089.

Jommi, C., Cinconze, E., Demattè, L., Guseo, R., Mortarino, C., Nosari, I., Pase, D., Roggeri, A., Roggeri, D., and Joppi, R. (2016). "The C-ToBIA model to estimate the budget impact of emerging drugs in horizon scanning". Manuscript submitted for publication.

Kim, T., Hong, J., and Koo, H. (2013). "Forecasting diffusion of innovative technology at pre-launch: A survey-based method". *Industrial Management and Data Systems*, 113, pp. 800–816.

Lee, H., Kim, S. G., Park, H., and Kang, P. (2014). "Pre-launch new product demand forecasting using the Bass model: A statistical and machine learning-based approach". *Technological Forecasting and Social Change*, 86, pp. 49–64.

Seber, G. A. F. and Wild, C. J. (1989). *Nonlinear regression*. Wiley: New York.

Wright, M. and Stern, P. (2015). "Forecasting new product trial with analogous series". *Journal of Business Research*, 68, pp. 1732–1738.

Technical Report Series
Department of Statistical Sciences, University of Padua

You may order paper copies of the working papers by emailing wp@stat.unipd.it
Most of the working papers can also be found at the following url: http://wp.stat.unipd.it

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI SCIENZE STATISTICHE

Department of Statistical Sciences
University of Padua
Italy