

Об одном алгоритме восстановления функции по разным функционалам для прогнозирования редких событий в экономике

Ю.А. Кораблев

Финансовый университет, Москва, Россия

АННОТАЦИЯ

Цель исследования — восстановление некоторых параметров функционала с использованием кубических сплайнов для дальнейшего прогнозирования редких событий в финансах и экономике. Рассмотрен математический **метод** восстановления неизвестной функции по многим разным функционалам, таким как значение функции, значение ее первой производной, второй производной, а также определенного интеграла на некотором промежутке. Причем все наблюдения могут происходить с погрешностью. Поэтому автор применил метод восстановления функции по разным функционалам, наблюдаемым с погрешностью. Восстановление функции осуществляется в виде кубического сплайна, имеющего представление через значения и вторые производные (value-second derivative representation). Оптимизационная задача заключается в минимизации сразу нескольких сумм квадратов отклонения: для обычных значений, для первых производных, для вторых производных, для интегралов, а также штрафа на нелинейность. Для большей гибкости введены весовые коэффициенты как для каждой группы наблюдений, так и для каждого индивидуального наблюдения в отдельности. Показано, как рассчитывается каждый отдельный функционал. Представлена компактная форма оптимизационной задачи через матричные операции. Подробно показано, как заполняется каждая соответствующая матрица. В приложении представлена программная реализация метода на языке R в виде функции `FunctionalSmoothingSpline`. Приведены примеры использования метода для анализа и прогнозирования редких (дискретных) событий в экономике. Представлены формулы расчета оценки кросс-валидации $CV(\alpha)$ для автоматической процедуры определения параметра сглаживания α из данных в нашей задаче восстановления функции по многим функционалам. Сделан **вывод**, что представленный метод позволяет анализировать и прогнозировать редкие события, что позволит подготовиться к ним, получить из этого определенную выгоду или уменьшить возможные риски или убытки.

Ключевые слова: редкие события; прогноз; анализ событий; восстановление по функционалам; сглаживающий сплайн; штраф на шероховатость; R; `FunctionalSmoothingSpline`; кросс-валидация

Для цитирования: Кораблев Ю.А. Об одном алгоритме восстановления функции по разным функционалам для прогнозирования редких событий в экономике. *Финансы: теория и практика*. 2022;26(3):196–225. DOI: 10.26794/2587-5671-2022-26-3-196-225

An Algorithm for Restoring a Function from Different Functionals for Predicting Rare Events in the Economy

Yu.A. Korablev

Financial University, Russia, Moscow

ABSTRACT

This paper aims to restore some parameters of functionals using cubic splines to forecast rare events in finance and economics. The article considers the mathematical method for recovering an unknown function from many different functionals, such as the value of a function, the value of its first derivative, second derivative, as well as a definite integral over a certain interval. Moreover, all observations can occur with an error. Therefore, the author uses a method of recovering a function from different functionals observed with an error. The function is restored in the form of a cubic spline, which has a value-second derivative representation. The optimization problem consists in minimizing several sums of squares of the deviation at once, for ordinary values, for the first derivatives, for the second derivatives, for integrals, and for roughness penalty. For greater flexibility, weights have been introduced both for each group of observations

and for each individual observation separately. The article shows in detail how the elements of each corresponding matrix are filled in. The appendix provides an implementation of the method as a `FunctionalSmoothingSpline` function in R language. Examples of using the method for the analysis and forecasting of rare (discrete) events in the economy are given. Formulas for calculating the cross-validation score $CV(\alpha)$ for the automatic procedure for determining the smoothing parameter α from the data in our problem of recovering a function by many functionals are shown. The paper concludes that the presented method makes it possible to analyze and predict rare events, which will allow you to prepare for such future events, get some benefit from this, or reduce possible risks or losses.

Keywords: rare events; forecast; event analysis; recovery by functionals; smoothing spline; roughness penalty; R; `FunctionalSmoothingSpline`; cross-validation

For citation: Korablev Yu.A. An algorithm for restoring a function from different functionals for predicting rare events in the economy. *Finance: Theory and Practice*. 2022;26(3):196-225. DOI: 10.26794/2587-5671-2022-26-3-196-225

1. ВВЕДЕНИЕ

Редкие события в экономике имеют особый интерес. Сама редкость событий уже с точки зрения теории информации делает такие события более значимыми [1]. В то же время события могут иметь значительные дорогостоящие последствия и способность их предсказания является актуальной задачей. В логистике, например, большой интерес вызывает предсказание редкого/прерывистого спроса (*intermittent demand forecast*), когда спрос возникает через большое количество интервалов, на которых спрос отсутствовал. Конечно же, абсолютно случайные события невозможно предсказать, но если в появлении событий есть какая-то закономерность, то предсказание возможно. Наиболее популярные методы Кростона [2] и Виллемейна [3], а также множество их модификаций способны найти статистические закономерности. В последних обзорах [4, 5] проанализированы существующие публикации, посвященные подобным методам прогноза прерывистого спроса. Но все равно во всех методах происходит лишь статистический анализ в той или иной степени. Из имеющихся данных (прерывистого спроса) определяются либо параметры распределений, либо значения переходных вероятностей простеньких моделей марковского процесса. Подобные подходы если и могут дать ожидаемое количество событий за интервал времени, то не способны дать прогноз момента времени возникновения конкретного события (дается оценка вероятности события в следующем временном периоде, что является неким статистическим «шаманизмом»).

Автором, в отличие от других исследователей, предложен подход [6], базирующийся на рассмотрении внутренних процессов, приводящих к появлению событий. Так, тот же редкий спрос следует анализировать с точки зрения процесса потребления, происходящего на стороне неподконтрольного нам клиента. Оказывается, что не составит труда по дискретным данным восстановить скорость

(интенсивность) потребления продукции у конкретного клиента. Конечно же, предварительно данные о событиях (покупках) надо разделить по разным выборкам в зависимости от источников (клиентов), где они образованы, что может быть не всегда возможно из-за несовершенства методики сбора данных. Рассматривая данные в выборке как последовательность интегралов от скорости потребления, происходит восстановление самой этой неизвестной функции скорости потребления с помощью методов восстановления по функционалам. Данный подход можно применить не только при анализе прерывистого спроса, а в любой области, где происходят процессы, схожие с опустошением (емкости) или же накоплением некоторого возмущения до определенного уровня, после чего оно сбрасывается до исходного уровня (этот метод автор назвал «емкостным методом»). В финансовой сфере данный подход также может найти применение, например, когда клиент периодически обращается за поддержкой.

Рассуждая аналогичным образом, можно задуматься, какие еще процессы могут существовать в экономике, которые порождают редкие события. Обращаю внимание, что нас не интересуют абсолютно случайные события, которые в принципе невозможно (или на данный момент не представляется возможным) предсказать, нас интересуют события, которые порождаются некоторым процессом, который мы могли бы сформулировать и потом самостоятельно воспроизвести.

Тут стоит вспомнить, а что же такое сама случайность. Случайность — это лишь мера неопределенности, мера незнания, абстракция, введенная для того, чтобы компактно обозначить все множество неизвестных исследователю факторов. Конечно же, благодаря повсеместному изучению статистики и теории вероятности в высших учебных заведениях и уже в школах, понятие случайности и вероятности стало чуть ли не физическим явлением. В то же время существует философская концепция

о космическом детерминизме, когда во вселенной все предопределено и некий мудрец, зная положение и скорость всех частиц в одно время, может предсказать их положение в любое другое время. Этой концепции противопоставляется принцип неопределенности Гейзенберга, когда в квантовой механике невозможно одновременно точно определить как положение, так и импульс частиц. Мы не будем спорить ни с той, ни с другой концепцией и рассуждать о том, существует истинная случайность или нет. Вышесказанное было нужно лишь для того, чтобы старшие коллеги, которые проработали многие десятилетия в области статистических исследований, взглянули на события или наблюдения не как на статистическую выборку, а попытались проникнуть в глубь каждого события и задумались, а какова причина появления этого события, каков механизм его образования. Взглянули на потоки событий не как на случайные потоки, а как на результат работы некоторого механизма образования этих событий, или даже множества механизмов, события от которых перемешиваются при неудачной реализации сбора данных.

Как было сказано выше, многие события в экономике связаны с процессами потребления (или накопления воздействия). Так как мы знаем, как они формируются, то потоки событий уже не случайные, эти знания вносят нужную нам информацию и позволяют более качественно их анализировать. Если не перемешивать события от разных источников, то в простейшем случае получается восстановить скорость потребления на стороне клиента, после чего построить модель закономерности и произвести расчеты следующего момента потребления. Для этого достаточно смотреть на значения объемов покупок как на интегралы. Однако в более сложном случае требуется привлекать и другую имеющуюся информацию. Данное исследование посвящено тому, как производить восстановление неизвестных параметров процесса, если на выбор будут доступны разные по характеру данные, такие как значения в определенный момент времени, значения первой и второй производной в некоторые моменты времени, значения определенных интегралов на определенных периодах времени. Причем эти разные по характеру данные могут быть доступны как все вместе, так и может быть доступно что-то одно, в то же время объемы выборок входных данных разных характеристик могут не совпадать (или равны нулю, если таких данных нет).

2. ВОССТАНОВЛЕНИЕ ФУНКЦИИ ПО РАЗНЫМ ФУНКЦИОНАЛАМ

2.1. Оптимизационная задача

Требуется восстановить динамически изменяющееся значение некоторого параметра процесса по имеющимся данным. Предполагаем, что данный параметр изменяется непрерывно, как некоторая неизвестная функция $f(t)$. То есть $f(t)$ —

искомая функция.

Пусть будут известны следующие данные.

$$y_i = f(t_i) + \varepsilon_i, \quad i = 1, \dots, n_f;$$

$$\dot{y}_j = f'(t_j) + \varepsilon_j, \quad j = 1, \dots, n_{df};$$

$$y_l'' = f''(t_l) + \varepsilon_l'', \quad l = 1, \dots, n_{d^2f};$$

$$Y_u = \int_{t_u^a}^{t_u^b} f(t) dt + \varepsilon_u^{\text{int}}, \quad u = 1, \dots, n_{\text{int}},$$

где t_i, t_j, t_l — моменты времени наблюдений за значениями неизвестной функции $f(t)$, ее первой и второй производной; t_u^a и t_u^b — нижний и верхний диапазон интегрирования у соответствующего наблюдения за интегралом; $\varepsilon_i, \varepsilon_j, \varepsilon_l'', \varepsilon_u^{\text{int}}$ — погрешности наблюдений у значений, первой производной, второй производной и интегралов соответственно (с нулевым математическим ожиданием, дисперсии могут быть разными); $n_f, n_{df}, n_{d^2f}, n_{\text{int}}$ — соответственно количество наблюдений значений, первых производных, вторых производных, интегралов искомой функции.

Конечно же, точно восстановить исходную функцию не получится, имеется бесконечное количество способов провести график функции таким образом, чтобы соответствующие значения у этой функции соответствовали заданным значениям в указанных точках. Можно приближенно восстановить функцию, налагая определенные ограничения. Например, можно сказать, что будем восстанавливать кусочками полиномов (сплайнами) определенной степени (третья степень самая распространенная). Будем налагать ограничения на гибкость (шероховатость, roughness) функции.

Подобный класс задач называют задачами коллокации, обратными задачами, задачами аппроксимации (можно рассматривать как синонимы в данном контексте). Считается (в англоязычной литературе), что основополагающей работой по приближенному восстановлению функций, является работа G. Kimeldorf и G. Wahba [7]. На текущий момент данная работа процитирована 797 раз

в международных базах цитирования. Изучение всего списка цитирующих работ и ознакомление более чем со 100 полнотекстовыми публикациями из этого списка (основываясь на аннотациях) показало отсутствие готового решения той задачи, которая решается в данной статье (тем более в том виде, в котором она представлена автором). Почему-то очень многие работы ограничиваются интерполяционными сплайнами, когда погрешности не учитываются, используются лишь интерполяционные условия в виде точных равенств. В некоторых работах можно встретить своеобразное решение лишь на основе только одного функционала, а не многих, и в основном опять с помощью интерполяционных сплайнов [8, 9].

Решение, представленное автором, основано на великолепно написанной работе P. J. Green и B. W. Silverman [10] и базируется на кубических сглаживающих сплайнах, имеющих представление в виде значений и вторых производных (value-second derivative representation). Для восстановления функции по многим функционалам будем минимизировать сразу несколько сумм квадратов отклонения и штрафа на нелинейность

$$S(f) = \sum_{i=1}^{n_f} (y_i - f(t_i))^2 + \sum_{j=1}^{n_{df}} (y'_j - f'(t_j))^2 + \sum_{l=1}^{n_{d^2f}} (y''_l - f''(t_l))^2 + \sum_{u=1}^{n_{int}} \left(Y_u - \int_{t_u^a}^{t_u^b} f(t) dt \right)^2 + \alpha \int_{t_{start}}^{t_{end}} (f''(t))^2 dt, \quad (1)$$

где последнее слагаемое — штраф на нелинейность; α — коэффициент сглаживания (регуляризации); t_{start} и t_{end} — соответственно границы, в которых происходит восстановление функции.

Так как величины, в которых измеряются значения, производные и интегралы, могут сильно отличаться друг от друга, добавим соответствующие коэффициенты, чтобы можно было увеличивать вес каждой группы наблюдений. Дополнительно добавим возможность изменять вес каждого отдельного наблюдения. В результате оптимизационная задача примет следующий вид:

$$S(f) = \sum_{i=1}^{n_f} w_i^f (y_i - f(t_i))^2 + \mu \sum_{j=1}^{n_{df}} w_j^{df} (y'_j - f'(t_j))^2 + \nu \sum_{l=1}^{n_{d^2f}} w_l^{d^2f} (y''_l - f''(t_l))^2 + \Psi \sum_{u=1}^{n_{int}} w_u^{int} \left(Y_u - \int_{t_u^a}^{t_u^b} f(t) dt \right)^2 + \alpha \int_{t_{start}}^{t_{end}} (f''(t))^2 dt, \quad (2)$$

где w_i^f , w_j^{df} , $w_l^{d^2f}$, w_u^{int} — индивидуальные веса соответствующих групп наблюдений;

μ — вес всей группы наблюдений первых производных; ν — вес всей группы наблюдений вторых производных; Ψ — вес всей группы наблюдений интегралов. Заметим, что отсутствует весовой коэффициент для всей группы наблюдений обычных значений, т.е. он предполагается равным единице, а все остальные коэффициенты тогда показывают вес по сравнению с этой первой группой наблюдений. Также заметим, что для изменения группового веса можно было бы пропорционально изменять все индивидуальные веса, однако это не очень удобно, поэтому будем использовать как индивидуальные, так и групповые веса.

Далее мы будем восстанавливать неизвестную функцию $f(t)$ в виде кубического сплайна $g(t)$ (сочленения кусочков полиномов третьей степени).

2.2. Форма сплайна

Вместо обычного представления полиномов с 4 неизвестными коэффициентами a_0, a_1, a_2, a_3 для каждого кусочка сплайна, определенного между узлами s_k и s_{k+1}

$$g(t) = a_0 + a_1(t - s_k) + a_2(t - s_k)^2 + a_3(t - s_k)^3, \quad s_k \leq t \leq s_{k+1}$$

мы будем использовать более удобное представление через значения сплайна $g_k = g(s_k)$ и значения его второй производной $\gamma_k = g''(s_k)$ на концах каждого интервала (value-second derivative representation) [10, с. 12, 22, 23]

$$g(t) = \frac{(t-s_k)g_{k+1} + (s_{k+1}-t)g_k}{s_{k+1}-s_k} - \frac{1}{6}(t-s_k)(s_{k+1}-t) \left[\left(1 + \frac{t-s_k}{s_{k+1}-s_k}\right)\gamma_{k+1} + \left(1 + \frac{s_{k+1}-t}{s_{k+1}-s_k}\right)\gamma_k \right], \quad (3)$$

$$s_k \leq t \leq s_{k+1}.$$

Как и раньше, каждый кусочек сплайна определяется 4 неизвестными $g_k, g_{k+1}, \gamma_k, \gamma_{k+1}$, но так как конец одного кусочка сплайна является началом следующего кусочка, то достаточно определить только два параметра g_k, γ_k для каждого узла s_k (отметим, что параметры g_k, γ_k содержат больше физического смысла, нежели параметры a_0, a_1, a_2, a_3). Для того чтобы определить сплайн во всех m узлах $s_1 < s_2 < \dots < s_m$ (количество узлов m обычно задается априори исследователем) необходимо задать вектор значений $g = (g_1, \dots, g_m)^T$ и вектор вторых производных $\gamma = (\gamma_2, \dots, \gamma_{m-1})^T$ (вторая производная на концах сплайна обращается в ноль $\gamma_1 = \gamma_m = 0$ — условия натурального сплайна).

Данная форма сплайна обеспечивает непрерывность $g(t)$ и ее второй производной $g''(t)$ в точках сочленения (узлах сплайна s_k). Однако для непрерывности первой производной в точках сочленения $g'(s_k-0) = g'(s_k+0)$ должна выполняться система $m-2$ уравнений

$$\frac{g_{k+1}-g_k}{s_{k+1}-s_k} - \frac{g_k-g_{k-1}}{s_k-s_{k-1}} = \frac{1}{6}(s_{k+1}-s_k)(\gamma_{k+1}+2\gamma_k) + \frac{1}{6}(s_k-s_{k-1})(2\gamma_k+\gamma_{k-1}), \quad (4)$$

$$k = 2, \dots, m-1;$$

или в матричном виде

$$Q^T g = R \gamma, \quad (5)$$

где Q — трехдиагональная матрица коэффициентов при неизвестных g_k размерностью $m \times (m-2)$ (записанных в столбец); R — трехдиагональная матрица коэффициентов при неизвестных γ размерностью $(m-2) \times (m-2)$ (тут $h_k = s_{k+1} - s_k$ расстояние между узлами для $k = 1, \dots, m-1$).

Q	2	3	...	$m-1$
1	h_1^{-1}	0	...	0
2	$-h_1^{-1} - h_2^{-1}$	h_2^{-1}	...	0
3	h_2^{-1}	$-h_2^{-1} - h_3^{-1}$	...	0
4	0	h_3^{-1}	...	0
...	...	...	...	...
$m-2$	0	0	...	h_{m-2}^{-1}
$m-1$	0	0	...	$-h_{m-2}^{-1} - h_{m-1}^{-1}$
m	0	0	...	h_{m-1}^{-1}

R	2	3	4	...	$m-1$
2	$(h_1 + h_2)/3$	$h_2/6$	0	...	0
3	$h_2/6$	$(h_2 + h_3)/3$	$h_3/6$	...	0
4	0	$h_3/6$	$(h_3 + h_4)/3$	...	0
5	0	0	$h_4/6$	...	0
...	...	...	...	...	...
$m-2$	0	0	...	...	$h_{m-2}/6$
$m-1$	0	0	...	...	$(h_{m-2} + h_{m-1})/3$

Вместо того, чтобы включать условия непрерывности первой производной $Q^T g = R\gamma$ в качестве системы ограничений в оптимизационную задачу, из этой системы уравнений выражают одну из неизвестных, например $\gamma = R^{-1}Q^T g$, подставляют ее и в дальнейшем решают оптимизационную задачу только с одной из неизвестных.

Штраф на гладкость (шероховатость) $\int_{s_1}^{s_m} (g''(t))^2 dt$ упрощается до операций с теми же матрицами, см. [10, с. 24–35]:

$$\int_{s_1}^{s_m} (g''(t))^2 dt = \gamma^T Q^T g = \gamma^T R \gamma = g^T [QR^{-1}Q^T] g = g^T K g, \quad (6)$$

где $K = QR^{-1}Q^T$ симметричная матрица размерностью $m \times m$.

2.3. Восстановление

В итоге имеем следующую задачу — определить параметры сплайна $g(t)$, минимизирующего

$$\begin{aligned} S(g) = & \sum_{i=1}^{n_f} w_i^f (y_i - g(t_i))^2 + \mu \sum_{j=1}^{n_{df}} w_j^{df} (y_j' - g'(t_j))^2 + \nu \sum_{l=1}^{n_{d^2f}} w_l^{d^2f} (y_l'' - g''(t_l))^2 \\ & + \psi \sum_{u=1}^{n_{int}} w_u^{int} \left(Y_u - \int_{t_u^a}^{t_u^b} g(t) dt \right)^2 + \alpha \int_{t_{start}}^{t_{end}} (g''(t))^2 dt, \end{aligned} \quad (7)$$

где сплайн $g(t)$ имеет форму (3).

Для упрощения записи удобно ввести обозначения $h_k = s_{k+1} - s_k$, $h_k^{-i} = t_i - s_k$, $h_k^{+i} = s_{k+1} - t_i$. Запись соответствующих функционалов будет следующей:

$$\begin{aligned} g(t_i) = & \frac{h_k^{-i}}{h_k} g_{k+1} + \frac{h_k^{+i}}{h_k} g_k - \frac{h_k^{-i} h_k^{+i} (h_k + h_k^{-i})}{6h_k} \gamma_{k+1} - \frac{h_k^{-i} h_k^{+i} (h_k + h_k^{+i})}{6h_k} \gamma_k, \\ & k : s_k \leq t_i < s_{k+1}; \end{aligned} \quad (8)$$

$$\begin{aligned} g'(t_j) = & \frac{g_{k+1}}{h_k} - \frac{g_k}{h_k} - \left(\frac{h_k}{6} - \frac{(h_k^{-j})^2}{2h_k} \right) \gamma_{k+1} + \left(\frac{h_k}{6} - \frac{(h_k^{+j})^2}{2h_k} \right) \gamma_k, \\ & k : s_k \leq t_j < s_{k+1}; \end{aligned} \quad (9)$$

$$g''(t_l) = \frac{h_k^{-l}}{h_k} \gamma_{k+1} + \frac{h_k^{+l}}{h_k} \gamma_k, \quad (10)$$

$$k : s_k \leq t_l < s_{k+1};$$

$$\int_{t_u^a}^{t_u^b} g(t) dt = \sum_{l=0}^L \int_{s_{k+l}}^{s_{k+l+1}} g(t) dt - \int_{s_k}^{t_u^a} g(t) dt - \int_{t_u^b}^{s_{k+L+1}} g(t) dt =$$

$$L : s_{k+L} < t_u^b \leq s_{k+L+1}, \quad (11)$$

$$k : s_k \leq t_u^a < s_{k+1}.$$

$$= \sum_{l=0}^L \left[\frac{h_{k+l}}{2} g_{k+l+1} + \frac{h_{k+l}}{2} g_{k+l} - \frac{h_{k+l}^3}{24} \gamma_{k+l+1} - \frac{h_{k+l}^3}{24} \gamma_{k+l} \right]$$

$$- \frac{(h_k^{-a})^2}{2h_k} g_{k+1} - \frac{h_k^2 - (h_k^{+a})^2}{2h_k} g_k - \frac{(h_k^{-a})^2 \left((h_k^{-a})^2 - 2h_k^2 \right)}{24h_k} \gamma_{k+1} + \frac{(h_k^{-a})^2 (h_k^{+a} + h_k)^2}{24h_k} \gamma_k$$

$$- \frac{h_{k+L}^2 - (h_{k+L}^{-b})^2}{2h_{k+L}} g_{k+L+1} - \frac{(h_{k+L}^{+b})^2}{2h_{k+L}} g_{k+L}$$

$$+ \frac{(h_{k+L}^{+b})^2 (h_{k+L}^{-b} + h_{k+L})^2}{24h_{k+L}} \gamma_{k+L+1} - \frac{(h_{k+L}^{+b})^2 \left((h_{k+L}^{+b})^2 - 2h_{k+L}^2 \right)}{24h_{k+L}} \gamma_{k+L}, \quad (12)$$

$$L : s_{k+L} < t_u^b \leq s_{k+L+1},$$

$$k : s_k \leq t_u^a < s_{k+1},$$

где в последнем выражении $h_k^{-a} = t_u^a - s_k$, $h_k^{+a} = s_{k+1} - t_u^a$, $h_k = s_{k+1} - s_k$, $h_{k+L}^{-b} = t_u^b - s_{k+L}$, $h_{k+L}^{+b} = s_{k+L+1} - t_u^b$, $h_{k+L} = s_{k+L+1} - s_{k+L}$.

Во всех этих выражениях в начале определяется, на какой интервал k выпало наблюдение. В самом последнем выражении для интеграла необходимо в начале определить, на какой интервал k выпал нижний предел интегрирования t_u^a и на какой интервал $k+L$ выпал верхний предел интегрирования t_u^b , где L — количество интервалов между ними (L может быть равно 0, если оба выпали на один интервал).

Все эти выражения имеют линейную форму относительно неизвестных параметров сплайна g_k и γ_k . Поэтому оптимизационную задачу (7) можно выразить в следующем матричном виде:

$$S(g) = (Y_f - V_f g + P_f \gamma)^T W_f (Y_f - V_f g + P_f \gamma) +$$

$$+ \mu (Y_{df} - V_{df} g + P_{df} \gamma)^T W_{df} (Y_{df} - V_{df} g + P_{df} \gamma) +$$

$$+ \nu (Y_{d^2f} - 0g + P_{d^2f} \gamma)^T W_{d^2f} (Y_{d^2f} - 0g + P_{d^2f} \gamma) +$$

$$+ \psi (Y_{int} - V_{int} g + P_{int} \gamma)^T W_{int} (Y_{int} - V_{int} g + P_{int} \gamma) +$$

$$+ \alpha g^T K g \rightarrow \min, \quad (13)$$

где Y_f , Y_{df} , Y_{d^2f} , Y_{int} — столбцы наблюдений; матрицы V_f , V_{df} , V_{int} — матрицы коэффициентов при неизвестных g_k ; P_f , P_{df} , P_{d^2f} , P_{int} — матрицы коэффициентов при неизвестных γ_k ; W_f , W_{df} , W_{d^2f} , W_{int} — диагональные матрицы весов.

V_f размерностью $n_f \times m$, ее каждая i -я строка выглядит как

1	...	$k-1$	k	$k+1$	$k+2$	...	m
0	...	0	h_k^{+i} / h_k	h_k^{-i} / h_k	0	...	0

P_f размерностью $n_f \times (m-2)$, ее каждая i -я строка выглядит как

2	...	$k-1$	k	$k+1$	$k+2$	...	$m-1$
0	...	0	$h_k^{-i} h_k^{+i} (h_k + h_k^{+i}) / 6h_k$	$h_k^{-i} h_k^{+i} (h_k + h_k^{-i}) / 6h_k$	0	...	0

V_{df} размерностью $n_{df} \times m$, ее каждая j -я строка выглядит как

1	...	$k-1$	k	$k+1$	$k+2$	...	m
0	...	0	$-1/h_k$	$1/h_k$	0	...	0

P_{df} размерностью $n_{df} \times (m-2)$, ее каждая j -я строка выглядит как

2	...	$k-1$	k	$k+1$	$k+2$	...	$m-1$
0	...	0	$-h_k / 6 + (h_k^{+j})^2 / 2h_k$	$h_k / 6 + (h_k^{-j})^2 / 2h_k$	0	...	0

P_{d^2f} размерностью $n_{d^2f} \times (m-2)$, ее каждая l -я строка выглядит как

2	...	$k-1$	k	$k+1$	$k+2$	...	$m-1$
0	...	0	$-h_k^{+l} / h_k$	$-h_k^{-l} / h_k$	0	...	0

V_{int} размерностью $n_{int} \times m$, ее каждая u -я строка заполняется следующим образом:

$$V_{u,k} = \frac{(h_k^{+a})^2}{2h_k}; V_{u,k+l} = \frac{h_{k+l-1} + h_{k+l}}{2}, l=1, \dots, L; V_{u,k+1} = V_{u,k+1} - \frac{(h_k^{-a})^2}{2h_k};$$

$$V_{u,k+L} = V_{u,k+L} - \frac{(h_{k+L}^{+b})^2}{2h_{k+L}}; V_{u,k+L+1} = \frac{(h_{k+L}^{-b})^2}{2h_{k+L}}.$$
(14)

Заметим, что в зависимости от L некоторые выражения могут измениться дважды (например, если $L=0$, то k -й элемент изменяется двумя выражениями $V_{i,k}$ и $V_{i,k+L}$). В случае $L>2$ строка u будет

$k-1$	k	$k+1$	$k+2$	$\dots k+l \dots$	$k+L$	$k+L+1$	$k+L+2$
0	$\frac{(h_k^{+a})^2}{2h_k}$	$\left(\frac{h_k + h_{k+1}}{2} - \frac{(h_k^{-a})^2}{2h_k} \right)$	$\frac{h_{k+1} + h_{k+2}}{2}$	$\frac{h_{k+l-1} + h_{k+l}}{2}$	$\left(\frac{h_{k+L-1} + h_{k+L}}{2} - \frac{(h_{k+L}^{+b})^2}{2h_{k+L}} \right)$	$\frac{(h_{k+L}^{-b})^2}{2h_{k+L}}$	0

В случае $L=0$ строка u будет

$k-1$	k	$k+1$	$k+2$
0	$\begin{pmatrix} \frac{(h_k^{+a})^2}{2h_k} \\ -\frac{(h_{k+L}^{+b})^2}{2h_{k+L}} \end{pmatrix}$	$\begin{pmatrix} \frac{(h_{k+L}^{-b})^2}{2h_{k+L}} \\ -\frac{(h_k^{-a})^2}{2h_k} \end{pmatrix}$	0

P_{int} размерностью $n_{int} \times (m-2)$, ее каждая u -я строка заполняется следующим образом:

$$\begin{aligned}
 P_{u,k} &= \frac{h_k^3}{24} - \frac{(h_k^{-a})^2 (h_k^{+a} + h_k)^2}{24h_k}; P_{u,k+l} = \frac{h_{k+l-1}^3 + h_{k+l}^3}{24}, l=1, \dots, L; \\
 P_{u,k+1} &= P_{u,k+1} + \frac{(h_k^{-a})^2 ((h_k^{-a})^2 - 2h_k^2)}{24h_k}; P_{u,k+L} = P_{u,k+L} + \frac{(h_{k+L}^{+b})^2 ((h_{k+L}^{+b})^2 - 2h_{k+L}^2)}{24h_{k+L}}; \\
 P_{u,k+L+1} &= \frac{h_{k+L}^3}{24} - \frac{(h_{k+L}^{+b})^2 (h_{k+L}^{-b} + h_{k+L})^2}{24h_{k+L}}.
 \end{aligned} \quad (15)$$

Тут тоже в зависимости от L некоторые значения могут поменяться несколько раз. В случае $L > 2$ строка u будет

$k-1$	k	$k+1$	$k+2$
0	$\begin{pmatrix} \frac{h_k^3}{24} - \frac{(h_k^{-a})^2 (h_k^{+a} + h_k)^2}{24h_k} \\ \frac{(h_{k+L}^{+b})^2 ((h_{k+L}^{+b})^2 - 2h_{k+L}^2)}{24h_{k+L}} \end{pmatrix}$	$\begin{pmatrix} \frac{h_k^3 + h_{k+1}^3}{24} + \frac{(h_k^{-a})^2 ((h_k^{-a})^2 - 2h_k^2)}{24h_k} \\ \frac{(h_{k+L}^{+b})^2 ((h_{k+L}^{+b})^2 - 2h_{k+L}^2)}{24h_{k+L}} \end{pmatrix}$	$\frac{h_{k+1}^3 + h_{k+2}^3}{24}$
$\dots k+l \dots$	$k+L$	$k+L+1$	$k+L+2$
$\frac{h_{k+l-1}^3 + h_{k+l}^3}{24}$	$\begin{pmatrix} \frac{h_{k+L-1}^3 + h_{k+L}^3}{24} + \frac{(h_{k+L}^{+b})^2 ((h_{k+L}^{+b})^2 - 2h_{k+L}^2)}{24h_{k+L}} \\ \frac{(h_{k+L}^{+b})^2 ((h_{k+L}^{+b})^2 - 2h_{k+L}^2)}{24h_{k+L}} \end{pmatrix}$	$\begin{pmatrix} \frac{h_{k+L}^3}{24} - \frac{(h_{k+L}^{+b})^2 (h_{k+L}^{-b} + h_{k+L})^2}{24h_{k+L}} \\ \frac{(h_{k+L}^{+b})^2 ((h_{k+L}^{+b})^2 - 2h_{k+L}^2)}{24h_{k+L}} \end{pmatrix}$	0

Далее, благодаря условиям непрерывности первой производной (5) $Q^T g = R\gamma$, позволяющим выразить $\gamma = R^{-1}Q^T g$, оптимизационную задачу (13) можно записать более компактно только через одну неизвестную g :

$$\begin{aligned}
 S(g) &= (Y_f - C_f g)^T W_f (Y_f - C_f g) + \mu (Y_{df} - C_{df} g)^T W_{df} (Y_{df} - C_{df} g) + \\
 &\quad + \nu (Y_{d^2f} - C_{d^2f} g)^T W_{d^2f} (Y_{d^2f} - C_{d^2f} g) + \\
 &\quad + \psi (Y_{int} - C_{int} g)^T W_{int} (Y_{int} - C_{int} g) + \alpha g^T K g \rightarrow \min,
 \end{aligned} \quad (16)$$

где $C_f = V_f - P_f R^{-1}Q^T$, $C_{df} = V_{df} - P_{df} R^{-1}Q^T$, $C_{d^2f} = 0 - P_{d^2f} R^{-1}Q^T$, $C_{int} = V_{int} - P_{int} R^{-1}Q^T$.

Наконец находим столбец неизвестных параметров g , приравнявая производную от $S(g)$ по g к нулю (правила для взятия производных в матричном виде

$$\frac{d(x^T b)}{dx} = b, \frac{d(bx)}{dx} = b^T, \frac{d(x^T Ax)}{dx} = (A + A^T)x = 2Ax, \text{ последнее верно, если } A - \text{ симметричная матрица).}$$

Выражение для искомых параметров g будет следующим:

$$g = \left(C_f^T W_f C_f + \mu C_{df}^T W_{df} C_{df} + \nu C_{d^2f}^T W_{d^2f} C_{d^2f} + \psi C_{int}^T W_{int} C_{int} + \alpha K \right)^{-1} \times \\ \times \left(C_f^T W_f Y_f + \mu C_{df}^T W_{df} Y_{df} + \nu C_{d^2f}^T W_{d^2f} Y_{d^2f} + \psi C_{int}^T W_{int} Y_{int} \right). \quad (17)$$

Зная g , рассчитывается $\gamma = R^{-1} Q^T g$, после чего уже можно строить сплайн $g(t)$ в произвольной точке t по выражению (3).

2.4. Выбор параметра сглаживания

Во всех подобных задачах восстановления функции (сглаживания) отдельно обсуждается выбор параметра сглаживания α . Процедуры автоматического выбора этого параметра для задачи восстановления по разным функционалам найти не удалось. Классические процедуры кросс-валидации или L -кривой не подойдут. Другие работы по автоматическому выбору параметра сглаживания для случая многих разных функционалов отсутствуют. Пришлось автору самостоятельно получить модифицированные формулы для расчета оценки кросс-валидации.

Напомним, что основная идея кросс-валидации заключается в том, чтобы подобрать такой параметр сглаживания α , чтобы восстановленная функция $g(t, \alpha)$ была эффективной с точки зрения предсказания, т.е. чтобы функция обладала наименьшей дисперсией при прогнозе следующих значений. Поэтому для расчета кросс-валидации исключают одно наблюдение, строят сплайн $g^{(-i)}(t, \alpha)$ и смотрят, с какой квадратичной ошибкой определяется это исключенное наблюдение, и так проделывают со всеми наблюдениями. В итоге оценка кросс-валидации дает некоторую оценку дисперсии наблюдений, как если бы они предсказывались по сплайну, оцененному по выборке с поэлементным исключением этих наблюдений. Формулы расчета такой оценки следующие (без промежуточных вычислений):

$$CV(\alpha) = n_f^{-1} \sum_{i=1}^{n_f} w_i^f \left(y_i - g^{(-i)}(t_i, \alpha) \right)^2 + n_{df}^{-1} \mu \sum_{j=1}^{n_{df}} w_j^{df} \left(y_j' - g^{(-j)}(t_j, \alpha) \right)^2 + \quad (18)$$

$$+ n_{d^2f}^{-1} \nu \sum_{l=1}^{n_{d^2f}} w_l^{d^2f} \left(y_l'' - g^{(-l)}(t_l, \alpha) \right)^2 + n_{int}^{-1} \psi \sum_{u=1}^{n_{int}} w_u^{int} \left(Y_u - \int_{t_u^a}^{t_u^b} g^{(-u)}(t, \alpha) dt \right)^2 = \\ = n_f^{-1} \sum_{i=1}^{n_f} w_i^f \left(\frac{y_i - g(t_i, \alpha)}{1 - \sum_{k=1}^m C_{ik}^f A_{ki}^f(\alpha)} \right)^2 + n_{df}^{-1} \sum_{j=1}^{n_{df}} w_j^{df} \left(\frac{y_j' - g'(t_j, \alpha)}{1 - \sum_{k=1}^m C_{jk}^{df} A_{kj}^{df}(\alpha)} \right)^2 + \quad (19)$$

$$+ n_{d^2f}^{-1} \sum_{l=1}^{n_{d^2f}} w_l^{d^2f} \left(\frac{y_l'' - g''(t_l, \alpha)}{1 - \sum_{k=1}^m C_{lk}^{d^2f} A_{kl}^{d^2f}(\alpha)} \right)^2 + n_{int}^{-1} \sum_{u=1}^{n_{int}} w_u^{int} \left(\frac{Y_u - \int_{t_u^a}^{t_u^b} g(t, \alpha) dt}{1 - \sum_{k=1}^m C_{uk}^{int} A_{ku}^{int}(\alpha)} \right)^2 = \\ = n_f^{-1} \sum_{i=1}^{n_f} w_i^f \left(\frac{y_i - \sum_{k=1}^m C_{ik}^f g_k}{1 - \sum_{k=1}^m C_{ik}^f A_{ki}^f(\alpha)} \right)^2 + n_{df}^{-1} \sum_{j=1}^{n_{df}} w_j^{df} \left(\frac{y_j' - \sum_{k=1}^m C_{jk}^{df} g_k}{1 - \sum_{k=1}^m C_{jk}^{df} A_{kj}^{df}(\alpha)} \right)^2 + \quad (20)$$

$$+ n_{d^2f}^{-1} \sum_{l=1}^{n_{d^2f}} w_l^{d^2f} \left(\frac{y_l'' - \sum_{k=1}^m C_{lk}^{d^2f} g_k}{1 - \sum_{k=1}^m C_{lk}^{d^2f} A_{kl}^{d^2f}(\alpha)} \right)^2 + n_{int}^{-1} \sum_{u=1}^{n_{int}} w_u^{int} \left(\frac{Y_u - \sum_{k=1}^m C_{uk}^{int} g_k}{1 - \sum_{k=1}^m C_{uk}^{int} A_{ku}^{int}(\alpha)} \right)^2.$$

Здесь верхние индексы $(-i), (-j), (-l), (-u)$ означают, что сплайн $g(t, \alpha)$ оценен по данным без этого конкретного наблюдения. Матрицы $C^f, C^{df}, C^{d^2f}, C^{int}$ — такие же, как и в формулах (16), (17). Матрицы $A^f(\alpha) = A(\alpha)C_f^T W_f$, $A^{df}(\alpha) = A(\alpha)\mu C_{df}^T W_{df}$, $A^{d^2f}(\alpha) = A(\alpha)v C_{d^2f}^T W_{d^2f}$, $A^{int}(\alpha) = A(\alpha)\psi C_{int}^T W_{int}$, где $A(\alpha) = (C_f^T W_f C_f + \mu C_{df}^T W_{df} C_{df} + v C_{d^2f}^T W_{d^2f} C_{d^2f} + \psi C_{int}^T W_{int} C_{int} + \alpha K)^{-1}$. Минимизация $CV(\alpha)$ любым известным способом относительно α дает искомое значение параметра сглаживания α .

Кросс валидация хорошо работает на обычных сплайнах. Однако у нас необычный случай, у нас происходит восстановление также по первым/вторым производным и интегралам. В итоге оценка кросс-валидации показывает не обычную дисперсию наблюдений, а дисперсию наблюдений как за значениями, так и за производными с интегралами (как вы, наверное, сами понимаете, происходит смешивание разных дисперсий, но при самом построении сплайна мы также смешивали квадраты ошибок разных наблюдений). Но проблема в том, что при исключении некоторых наблюдений может катастрофически сильно поменяться вид восстановленной функции. В описанных ниже примерах (особенно во втором) будет очевидно, что исключение любого наблюдения приведет к тому, что функция определится очень неточно, при этом рассчитывается ошибка прогноза именно для исключенного наблюдения. Когда каждое наблюдение несет крайне необходимую информацию для восстановления функции, исключение наблюдения из выборки приведет к очень и очень большим ошибкам. В этом случае кросс-валидация не подходит. Но если же у нас очень много наблюдений, одни наблюдения лишь статистически дополняют информацию других наблюдений, и исключение одного из наблюдений не приводит к значительным изменениям восстановленной функции, то метод кросс-валидации может быть хорошим решением для определения параметра сглаживания.

3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ НА ЯЗЫКЕ R

Описанный метод восстановления функции сразу по многим разным функционалам реализован на языке R в виде функции `FunctionalSmoothingSpline` (см. Приложение 1). Существующих готовых функций и пакетов в R или других языках, реализующих подобные возможности, обнаружить не удалось.

4. ПРИМЕР ИСПОЛЬЗОВАНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ РЕДКИХ СОБЫТИЙ В ЭКОНОМИКЕ

4.1. Прогнозирование будущих покупок клиента

Пусть есть некоторый клиент, который покупает продукцию у нас (например, обычный покупатель делает покупки в некоторой торговой сети, или же некоторый оптовик покупает продукцию на заводе изготовителе). Нам про клиента ничего не известно, кроме дат и объемов его покупок. Для простоты смоделируем процесс пополнения запасов как в классических моделях управления запасами. Пусть данные о покупках будут выглядеть следующим образом (табл. 1).

Таблица 1 / Table 1

Данные о покупках неподконтрольного нам клиента / Purchases data of a client not controlled by us

Дата / Date t_i	Объем / Volume y_i	Дата / Date t_i	Объем / Volume y_i	Дата / Date t_i	Объем / Volume y_i	Дата / Date t_i	Объем / Volume y_i	Дата / Date t_i	Объем / Volume y_i
03.01.2020	2170	06.06.2020	1976	09.10.2020	2093	10.04.2021	2257	07.08.2021	1968
25.01.2020	2281	19.06.2020	2205	24.10.2020	2141	28.04.2021	2189	19.08.2021	2136
22.02.2020	2242	03.07.2020	2096	12.11.2020	2273	12.05.2021	2026	02.09.2021	2145
11.03.2020	2206	17.07.2020	2125	10.12.2020	2217	24.05.2021	2072	20.09.2021	2235
26.03.2020	2142	29.07.2020	2034	31.12.2020	2218	04.06.2021	1983	07.10.2021	2186
12.04.2020	2210	09.08.2020	1980	21.01.2021	2252	16.06.2021	2059	22.10.2021	2141
30.04.2020	2215	21.08.2020	2098	18.02.2021	2211	30.06.2021	2146	09.11.2021	2256
14.05.2020	2102	05.09.2020	2222	09.03.2021	2218	14.07.2021	2082	07.12.2021	2264
26.05.2020	2115	23.09.2020	2191	24.03.2021	2137	27.07.2021	2177	29.12.2021	2241

Источник / Source: составлено автором / compiled by the author.

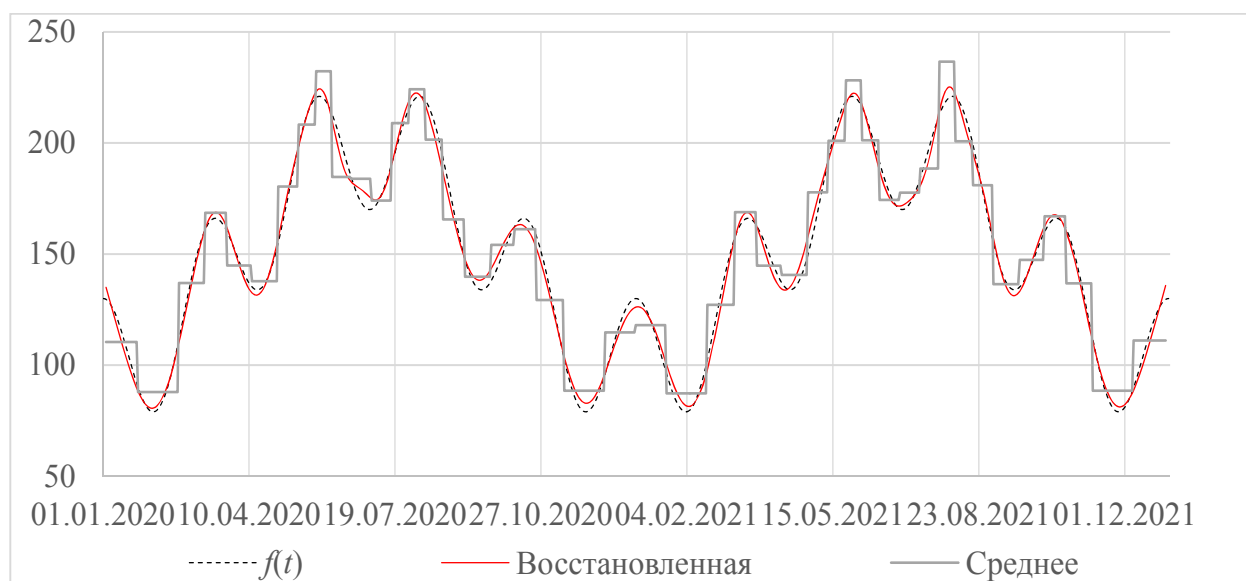


Рис. 1 / Fig. 1. Восстановленная интенсивность потребления продукции на стороне неподконтрольного клиента / The restored intensity of product consumption on the side of the uncontrolled customer

Источник / Source: составлено автором / compiled by the author.

Примечание / Note: сплошная линия — восстановленная интенсивность потребления; пунктирная линия — исходная функция потребления продукции; ступенчатая линия — средняя интенсивность $y_i / (t_{i+1} - t_i)$; площадь под каждой ступенькой означает объем покупки y_i / solid line — restored consumption intensity; dashed line — original consumption rate; stepped line — average intensity.

Воспринимая эти данные как последовательность интегралов от неизвестной функции интенсивности потребления за время от текущего события до следующего, восстанавливаем с помощью описанного выше математического метода эту неизвестную функцию (рис. 1). В Приложении 2 представлена реализация на языке R, показано, как подготовить данные из csv файла и как вызвать функцию FunctionalSmoothingSpline. Можно увидеть, что для текущего набора данных восстановить неизвестную функцию интенсивности потребления неподконтрольного нам клиента получилось очень хорошо.

Следующим шагом необходимо определить закономерность в выявленной функции, построить модель и провести экстраполяцию. Причем для этого могут использоваться любые известные методы. Саму зависимость восстановленной функции можно строить только от времени, а можно определять зависимость от некоторых известных внешних факторов. Здесь предполагается полная свобода и ответственность исследователя. В нашем случае будем строить модель и экстраполировать значения на будущее в виде суммы гармонических функций (при моделировании данных закладывалась гармоническая функция). Для этого хорошо подходит метод Куинна и Фернандеса (Quinn–Fernandes algorithm) [11, 12], который представляет исходные данные в виде суммы ограниченного числа гармонических функций. Результат такой экс-

траполяции представлен на рис. 2. Так как восстановленная функция содержала некоторые отклонения от истинной функции, заложенной при моделировании, то параметры экстраполированной функции определились с некоторой неточностью, в результате чего на некоторых участках экстраполированная функция заметно расходится с истинной.

Окончательным шагом является запуск самого процесса определения будущих событий. В нашем случае запускается вновь процесс потребления как в системах управления запасами, где в качестве функции потребления участвуют экстраполированные значения. Начиная с самого последнего наблюдения, можно определить, когда закончится запас у клиента и тем самым дать прогноз следующего обращения. Если из данных определить еще максимальный запас (или построить модель для объемов покупок), то получится спрогнозировать всю последовательность будущих событий. На рис. 2 на горизонтальной оси крестиками показан прогноз будущих событий. Причем первые несколько событий определяются с погрешностью всего в 0–2 дня, далее погрешность увеличивается до 6 дней, но потом опять уменьшается до 2 дней (и это при том, что время между событиями 15–28 дней). То есть первые события получается определить очень хорошо, но при увеличении горизонта планирования точность падает (в данном примере потом опять улучшается), что закономерно.



Рис. 2 / Fig. 2. Экстраполяция восстановленной функции и прогноз будущих событий / Extrapolation of the restored function and forecast of future events

Источник / Source: составлено автором / compiled by the author.

4.2. Определение скрытой динамики фирмы по сообщениям, содержащим преимущественно качественную информацию

Рассуждая, какие еще в экономике есть процессы образования событий, в первую очередь приходят в голову такие, которые можно так или иначе свести к процессам, похожим на процессы потребления или накопления возмущений (опустошения запаса или накопления возмущений до критического уровня). Напоминаю, что абсолютно случайные события нас не интересуют, там, где нет закономерности, нечего ее искать. Попробуйте задуматься, какие вам еще приходят на ум события, чей процесс образования отличен от процессов потребления или накопления воздействия. Не сразу, а спустя длительный период размышлений, получилось сформулировать процесс порождения редких событий в экономике, отличный от упомянутых, далее речь пойдет об одном таком процессе. Стоит сказать, что рассматриваемый далее процесс в большей мере демонстрирует возможность применения предлагаемого подхода для анализа редких событий, нежели решает конкретные практические задачи, хотя такие непременно найдутся (нас интересует возможность анализа редких событий, образованных процессами, отличными от процессов потребления).

Пусть есть некоторая неподконтрольная нам организация, от которой в определенные дискретные моменты времени поступают сигналы, несущие преимущественно качественную (а не количественную) информацию. Причем в каждом сигнале эта качественная информация может относиться к разным

характеристикам (не обязательно, что каждый сигнал сообщает обо всех характеристиках). Для примера представьте такую картину, когда от некоторой компании поступают следующие сигналы:

- Мы начали тонуть — надо срочно что-то делать!
- Мы все еще сильно тонем!
- Кажется, снижение начало замедляться.
- Мы начали всплывать.
- Хорошо идем, вот бы всегда так.
- О нет, мы вновь тонем!
- Мы прошли отметку невозврата...
- Мы на дне.

Что можно извлечь из такого набора сигналов? Можно догадаться, что события образованы внутри такой организации в результате наблюдения за некоторым внутренним показателем, который изменяется динамически со временем (возможно вследствие каких-то управляющих воздействий). Образование событий связано с операцией сравнения этого показателя с некоторыми критическими значениями. Если воспользоваться предложенной методикой, восстановить динамику ненаблюдаемого напрямую внутреннего показателя организации, построить модель изменения этой динамики, провести экстраполяцию и вновь запустить процесс образования событий, то возможно будет предсказать будущие события (и возможно даже управляющие воздействия).

Что же нам известно из подобного вида сигналов, что поможет восстановить динамику изменения ненаблюдаемого напрямую показателя. Оказывается, из имеющихся сигналов можно извлечь сле-

дующие данные: моменты времени как моменты получения сообщений (с возможной поправкой на задержку в получении); значения первой производной или положение точек экстремума; значения второй производной или положение точек перегиба; значения самого изменяющегося показателя в некоторых точках. Для большей полноты также предположим, что могут наблюдаться интегралы от функции изменения того же ненаблюдаемого показателя (отсутствует в описанном выше примере, но можно предположить, что если интересующий нас показатель указывает на финансовые средства на счетах, то начисленные проценты по этим счетам за некоторые периоды времени будут этими интегралами). Следует отметить, что все наблюдения могут быть неточными, причем неточными могут быть как сами моменты наблюдения, так и значения в этих наблюдениях. Однако погрешность в моментах наблюдения так или иначе может быть сведена к погрешности в значениях этих же наблюдений (например, если истинное положение экстремума чуть сдвинуто от момента наблюдения, то в этот момент наблюдения значение производной будет смещено относительно нуля).

Обозначим неизвестную искомую функцию скрытой динамики как $f(t)$, которую в дальнейшем будем восстанавливать по имеющимся данным в виде сплайна. Для простоты пусть функция $f(t)$ будет безразмерной и в начальный момент времени t_0 значение этой функции будет принято за исходное значение, а все остальные значения выражаются в процентах от этого исходного (т.е. полагаем, что $f(t_0) = 100$). Далее, для примера, пусть исходная неизвестная функция ведет себя так, как показано на рис. 3.

Изначально было сказано, что мы имеем дело с преимущественно качественными данными, но этим качественным данным не составит труда дать количественную оценку, причем достаточно дать приближенную оценку (или же значения могли быть изначально приближенно известны). Так, в событиях a, c, d, f говорим, что наблюдаются нулевые значения первых или вторых производных. Для событий b и e предположим, что нам приближенно известен тангенс угла наклона, для события g предположим (или знаем), что уровень невозврата начинается на 20%. Мы давали приближенную оценку значениям, а моменты времени событий наблюдаются и известны. В результате имеющиеся данные могут быть следующими (табл. 2).

Как можно заметить, объем выборок может сильно отличаться, для второй производной имеем только одно наблюдение. Реализуя описанный метод восстановления функции по функционалам (все коэффициенты μ, ν, ψ, α равны 1), из имеющихся данных

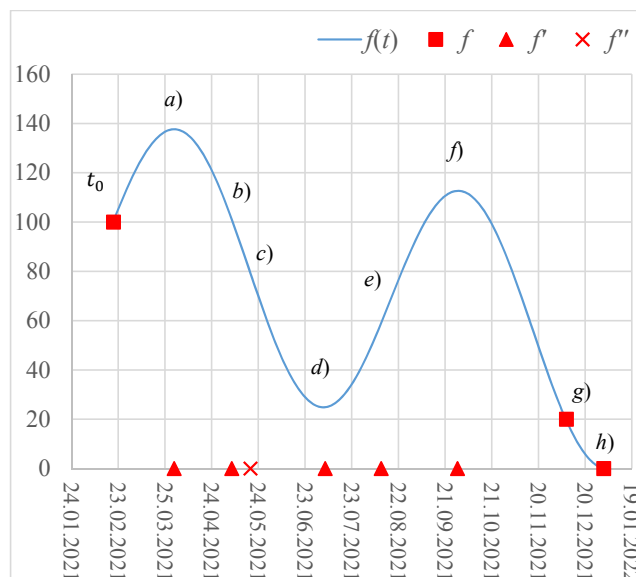


Рис. 3 / Fig. 3. Пример исходной функции $f(t)$ и исходных данных / An example of the initial function $f(t)$ and the initial data

Источник / Source: составлено автором / compiled by the author.

Примечание / Note: функция $f(t)$ безразмерная, показывает количество (%) от начального уровня (значение в начальный момент t_0 берется как исходное). Данные (события) значений функции отмечены на самом графике, производные отложены на горизонтальной оси и обозначают их позицию / Function $f(t)$ is dimensionless, shows the number (%) of the initial level (the value at the moment t_0 is taken as the initial). The function value data (events) are marked on the graph itself, the derivatives data are plotted on the horizontal axis and indicate their position.

получим следующий результат (рис. 4). Код на языке R представлен в Приложении 1. Задача оказалась плохо обусловленной, было недостаточно информации для хорошего восстановления функции (не было задано, что в самом начале функция должна была возрастать).

Если же добавить больше информации, например, в виде интегралов искомой функции, (рис. 5), (табл. 3), то восстановить исходную функцию получается намного точнее (рис. 6).

Конечно же восстановление все еще не идеальное, но это связано с нехваткой самой информации, а не с недостатком математического метода. Если добавить больше наблюдений, можно добиться очень хорошего восстановления. Заметим, что с помощью настройки весов как самих наблюдений, так и групп наблюдений, можно очень гибко настраивать, какой информации стоит больше уделять внимания, а какой меньше.

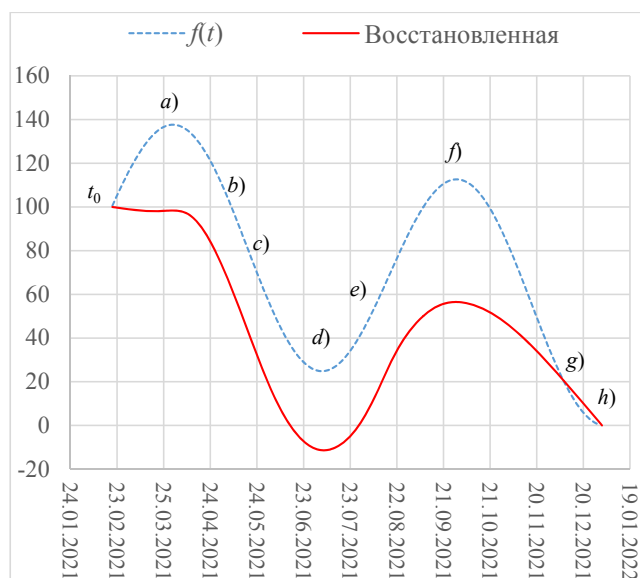
Дальнейшее прогнозирование будущих событий подчиняется все той же схеме, как и в предыдущем примере (в данном примере мы можем и не ждать никаких будущих событий).

Таблица 2 / Table 2

Имеющиеся приблизительные данные / Available approximate data

t_f	y_f	t_{df}	y_{df}	t_{d^2f}	y_{d^2f}
20.02.2021	100	31.03.2021	0	19.05.2021	0
08.12.2021	20	07.05.2021	-1,75	-	-
01.01.2022	0	06.07.2021	0	-	-
-	-	11.08.2021	1,55	-	-
-	-	29.09.2021	0	-	-

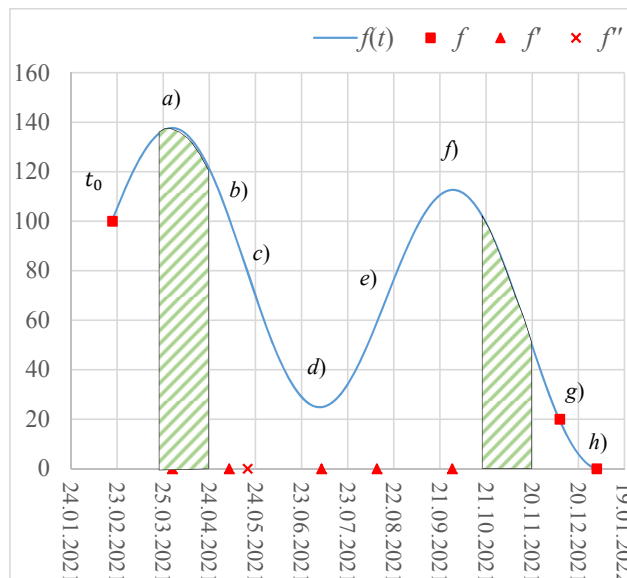
Источник / Source: составлено автором / compiled by the author.

Рис. 4 / Fig. 4. Плохо обусловленная задача.
Информации было недостаточно /
Poorly conditioned task. There was
not enough information

Источник / Source: составлено автором / compiled by the author.

5. СХЕМА ПОДХОДА
ДЛЯ ПРОГНОЗИРОВАНИЯ
РЕДКИХ СОБЫТИЙ

Еще раз отдельно приведем идею подхода прогнозирования редких событий. Если из редких событий получается восстановить параметры процесса (динамику), то дальнейшими действиями для прогнозирования редких событий будет выявление закономерности изменения этих параметров процесса (их динамики) со временем. Причем для этого следует задействовать всю возможную информацию, можно определить закономерность в зависимости от изменений внешних наблюдаемых факторов, таких как ВВП, инфляция, безработица и других факторов. Для этой цели могут использоваться любые существ-

Рис. 5 / Fig. 5. Исходная функция $f(t)$ и исходные
данные в виде событий с добавлением
информации об интегралах функции / The original
function $f(t)$ and the initial data with the
additional information about the integrals

Источник / Source: составлено автором / compiled by the author.

ующие математические методы. Целью этого этапа будет построение соответствующей модели изменения внутренних параметров процесса. На следующем шаге необходимо провести экстраполяцию параметров процесса на будущее. Если ранее строилась модель зависимости от внешних факторов, то предварительно нужно экстраполировать значения внешних факторов. Для этапа экстраполяции можно опять использовать любой известный математический метод. Наконец, если мы имеем представления о том, как будут изменяться в будущем параметры процесса образования событий, и мы можем воспроизвести механизм этого процесса, то не составит труда получить прогноз будущих событий, запустив с установленными параметрами сам этот процесс.

Таблица 3 / Table 3

Данные с добавлением информации об интегралах / Data with added information about integrals

t_f	y_f	t_{df}	y_{df}	t_{d^2f}	y_{d^2f}	t_{int}^a	t_{int}^b	Y_{int}
20.02.2021	100	31.03.2021	0	19.05.2021	0	25.03.2021	24.04.2021	4000
08.12.2021	20	07.05.2021	-1,75	-	-	21.10.2021	20.11.2021	2282
01.01.2022	0	06.07.2021	0	-	-	-	-	-
-	-	11.08.2021	1,55	-	-	-	-	-
-	-	29.09.2021	0	-	-	-	-	-

Источник / Source: составлено автором / compiled by the author.

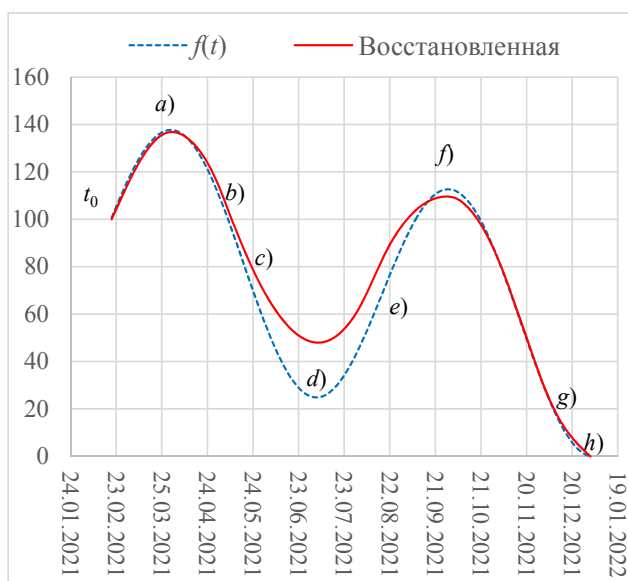


Рис. 6 / Fig. 6. Восстановление функции по данным с добавлением информации об интегралах / Restoration of a function from data with the additional information about integrals

Источник / Source: составлено автором / compiled by the author.

Схема получения прогноза будущих событий представлена на рис. 7.

Напрямую нельзя сравнить предлагаемый подход с другими методами. Дело в том, что ни один из существующих методов не может дать прогноз момента времени наступления будущего события. В одних методах дается прогноз возникновения заданного количества событий за определенное количество времени либо определяется лишь вероятность возникновения одного события в следующий интервал времени (статистические методы на основе пуассоновских и других потоков событий, методы анализа нерегулярного спроса). В других методах определяется вероятность возникновения события при условии наблюдения определенной последовательности среди наблюдаемых внешних факторов или лаговых переменных (всевозможные методы классификации). Более того, в описанном подходе предлагается строить модель и делать экстраполяцию восстановленных зависимостей внутренних параметров со временем, когда в других методах все оценки получаются статичными. Для сравнения предлагаемого подхода предсказания

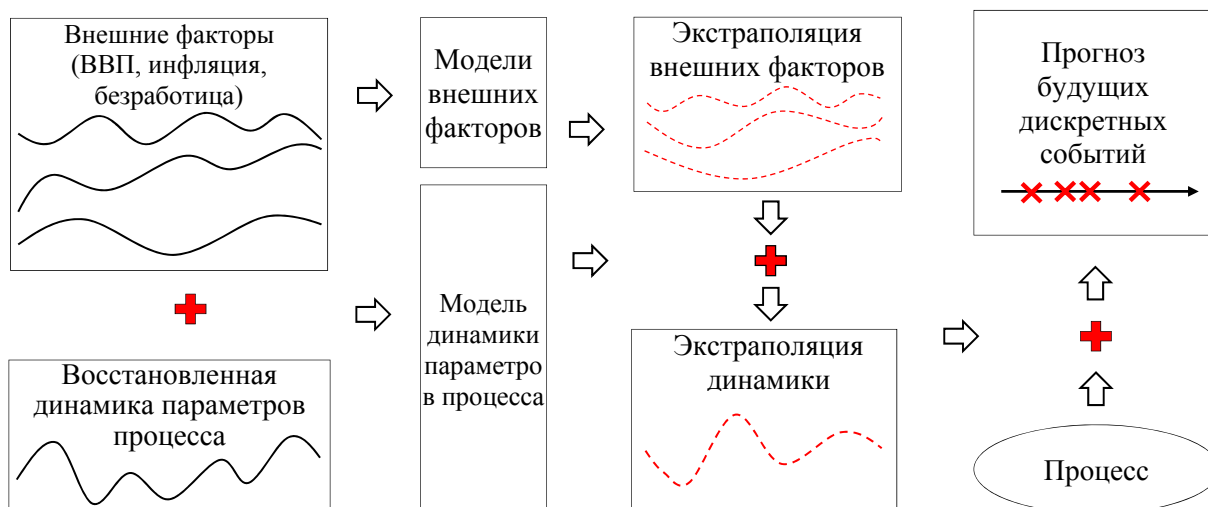


Рис. 7 / Fig. 7. Схема подхода для прогнозирования редких событий в экономике / Scheme of the approach for predicting rare events in the economy

Источник / Source: составлено автором / compiled by the author.

событий с другими методами придется адаптировать его под специфические задачи других методов. То есть с помощью метода надо дать прогноз сразу множеству будущих событий за интервал времени, причем сразу от множества источников (от разных клиентов). Такой смешанный прогноз группы событий можно будет сравнить с прогнозом других методов, которые работают с событиями, представленными в форме временных рядов. Однако надо учесть, что у предлагаемого подхода, помимо этапа восстановления динамики внутренних параметров процесса, существуют этапы выявления закономерности и экстраполяции динамики параметров процесса, осуществляемых любым известным математическим методом. В зависимости от того, какие методы для этого выберет исследователь, эффективность предлагаемого подхода может быть оценена по-разному. Для сравнения эффективности с другими методами необходимо проводить отдельное объемное исследование, а может даже и не одно.

ВЫВОДЫ

В результате данного исследования удалось разработать математический метод восстановления функции одновременно по многим разным функционалам с учетом погрешности в наблюдении этих функционалов. В *Приложении 1* представлена программная реализация метода на языке R. Благодаря описанному методу можно анализировать и предсказывать редкие события, которые порождаются некоторыми процессами. В случае, если это

процессы потребления, то достаточно рассматривать данные (покупки, кредиты и прочее) как интегралы и восстанавливать зависимости из этой последовательности интегралов. Если же это более сложные процессы, то дополнительно некоторые данные можно рассматривать как первые или вторые производные. Если получается восстановить динамику параметров процесса образования редких событий из имеющихся данных об этих событиях, то следующим шагом будет определение закономерности и экстраполяция этой динамики на будущее. Причем на этом этапе исследователь ничем не ограничен и может использовать любой подходящий математический метод. После этапа экстраполяции можно запустить сам процесс образования событий с установленными значениями внутренних параметров и получить прогноз будущих событий в экономике. Описанный подход можно применять в различных областях, например, анализируя данные о стрижке некоторого постоянного клиента в парикмахерской, можно восстановить функцию скорости накопления желания подстричься [13], а анализируя исторические данные русско-турецких войн, можно получить скорость нарастания разногласий или скорость подготовки к очередной войне [14] (последнее имеет больше демонстративный характер). Анализ и прогнозирование редких событий имеет очень важное значение. Это позволит подготовиться к таким событиям, получить из этого определенную выгоду или уменьшить возможные риски или убытки.

БЛАГОДАРНОСТИ

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-010-00154. Финансовый университет, Москва, Россия.

ACKNOWLEDGEMENTS

The study was funded by the Russian Foundation for Basic Research (RFBR), project No 19-010-00154. Financial University, Moscow, Russia.

СПИСОК ИСТОЧНИКОВ

1. Шеннон К. Работы по теории информации и кибернетике. Пер. с англ. М.: Изд. иностр. лит.; 1963. 829 с.
2. Croston J. D. Forecasting and stock control for intermittent demands. *Operational Research Quarterly*. 1972;23(3):289–303. DOI: 10.1057/jors.1972.50
3. Willemain T. R., Smart C. N., Schwarz H. F. A new approach to forecasting intermittent demand for service parts inventories. *International Journal of Forecasting*. 2004;20(3):375–387. DOI: 10.1016/S 0169-2070(03)00013-X
4. Kaya G. O., Sahin M., Demirel O. F. Intermittent demand forecasting: A guideline for method selection. *Sādhanā: Academy Proceedings in Engineering Sciences*. 2020;45(1):51. DOI: 10.1007/s12046-020-1285-8
5. Pinçe Ç., Turrini L., Meissner J. Intermittent demand forecasting for spare parts: A critical review. *Omega*. 2021;105:102513. DOI: 10.1016/j.omega.2021.102513
6. Кораблев Ю. А. Метод восстановления функции по интегралам для анализа и прогнозирования редких событий в экономике. *Экономика и математические методы*. 2020;56(3):113–124. DOI: 10.31857/S 042473880010485-2

7. Kimeldorf G., Wahba G. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*. 1971;33(1):82–95. DOI: 10.1016/0022-247X(71)90184-3
8. Lee C.H. A phase space spline smoother for fitting trajectories. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*. 2004;34(1):346–356. DOI: 10.1109/tsmcb.2003.817027
9. Xian J., Li Y., Lin W. Reconstruction of signal from samples of its integral in spline subspaces. In: Bubak M., van Albada G.D., Sloot P.M.A., Dongarra J., eds. Int. conf. on computational science: Computational science (ICCS 2004). Berlin, Heidelberg: Springer-Verlag; 2004:574–577. (Lecture Notes in Computer Science. Vol. 3037). DOI: 10.1007/978-3-540-24687-9_75
10. Green P.J., Silverman B.W. Nonparametric regression and generalized linear models. A roughness penalty approach. Boca Raton, London: Chapman & Hall/CRC; 1994. 194 p. (Monographs on Statistics and Applied Probability. No. 58).
11. Quinn B.G., Fernandes J.M. A fast efficient technique for the estimation of frequency. *Biometrika*. 1991;78(3):489–497. DOI: 10.1093/biomet/78.3.489
12. Quinn B.G., Hannan E.J. The estimation and tracking of frequency. Cambridge, New York: Cambridge University Press; 2001. 278 p. (Cambridge Series in Statistical and Probabilistic Mathematics. No. 9).
13. Кораблев Ю.А., Голованова П.С., Кострица Т.А. Емкостный метод анализа редких событий в сфере услуг. *Экономическая наука современной России*. 2020;(3):132–142. DOI: 10.33293/1609-1442-2020-3(90)-132-142
14. Кораблев Ю.А., Голованова П.С., Кострица Т.А. Использование емкостного метода для анализа исторических событий. *KANT*. 2021;(1):27–32. DOI: 10.24923/2222-243X.2021-38.6

REFERENCES

1. Shannon C. Works on information theory and cybernetics. Transl. from Eng. Moscow: Foreign Literature Publ.; 1963. 258 p. (In Russ.).
2. Croston J.D. Forecasting and stock control for intermittent demands. *Operational Research Quarterly*. 1972;23(3):289–303. DOI: 10.1057/jors.1972.50
3. Willemain T.R., Smart C.N., Schwarz H.F. A new approach to forecasting intermittent demand for service parts inventories. *International Journal of Forecasting*. 2004;20(3):375–387. DOI: 10.1016/S 0169-2070(03)00013-X
4. Kaya G.O., Sahin M., Demirel O.F. Intermittent demand forecasting: A guideline for method selection. *Sādhanā: Academy Proceedings in Engineering Sciences*. 2020;45(1):51. DOI: 10.1007/s12046-020-1285-8
5. Pinçe Ç., Turrini L., Meissner J. Intermittent demand forecasting for spare parts: A critical review. *Omega*. 2021;105:102513. DOI: 10.1016/j.omega.2021.102513
6. Korablev Yu.A. The function restoration method by integrals for analysis and forecasting of rare events in the economy. *Ekonomika i matematicheskie metody = Economics and Mathematical Methods*. 2020;56(3):113–124. (In Russ.). DOI: 10.31857/S 042473880010485-2
7. Kimeldorf G., Wahba G. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*. 1971;33(1):82–95. DOI: 10.1016/0022-247X(71)90184-3
8. Lee C.H. A phase space spline smoother for fitting trajectories. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*. 2004;34(1):346–356. DOI: 10.1109/tsmcb.2003.817027
9. Xian J., Li Y., Lin W. Reconstruction of signal from samples of its integral in spline subspaces. In: Bubak M., van Albada G.D., Sloot P.M.A., Dongarra J., eds. Int. conf. on computational science: Computational science (ICCS 2004). Berlin, Heidelberg: Springer-Verlag; 2004:574–577. (Lecture Notes in Computer Science. Vol. 3037). DOI: 10.1007/978-3-540-24687-9_75
10. Green P.J., Silverman B.W. Nonparametric regression and generalized linear models. A roughness penalty approach. Boca Raton, London: Chapman & Hall/CRC; 1994. 194 p. (Monographs on Statistics and Applied Probability. No. 58).
11. Quinn B.G., Fernandes J.M. A fast efficient technique for the estimation of frequency. *Biometrika*. 1991;78(3):489–497. DOI: 10.1093/biomet/78.3.489
12. Quinn B.G., Hannan E.J. The estimation and tracking of frequency. Cambridge, New York: Cambridge University Press; 2001. 278 p. (Cambridge Series in Statistical and Probabilistic Mathematics. No. 9).
13. Korablev Yu.A., Golovanova P.S., Kostritsa T.A. Capacity method of rare events analysis in the area of services. *Ekonomicheskaya nauka sovremennoi Rossii = Economics of Contemporary Russia*. 2020;(3):132–142. (In Russ.). DOI: 10.33293/1609-1442-2020-3(90)-132-142
14. Korablev Yu.A., Golovanova P.S., Kostritsa T.A. Using the capacity method to analyze historical events. *KANT*. 2021;(1):27–32. (In Russ.). DOI: 10.24923/2222-243X.2021-38.6

Реализация на языке R

Восстановление функции по функционалам реализовано в виде функции `FunctionalSmoothingSpline`, на вход которой подаются имеющиеся данные.

```
FunctionalSmoothingSpline = function (
  t_f = NULL,      # array of observation moments
  values_f = NULL, # array of observation values
  weights_f = NULL, # array of weights
  t_df = NULL,     # array of first derivative moments
  values_df = NULL, # array of first derivative values
  weights_df = NULL, # array of first derivative weights
  coef_df = 1,     # coefficient of first derivative sum of squares
  t_d2f = NULL,    # array of second derivative moments
  values_d2f = NULL, # array of second derivative values
  weights_d2f = NULL, # array of second derivative weights
  coef_d2f = 1,    # coefficient of second derivative sum of squares
  t_int_a = NULL,  # array of intervals start moments
  t_int_b = NULL,  # array of intervals end moments
  values_int = NULL, # array of interval values
  weights_int = NULL, # array of interval weights
  coef_int = 1,    # coefficient of integral sum of squares
  knots = NULL,    # knots
  knots_number = NULL, # number of knots
  alpha = 1,       # smoothing parameter
  x = NULL,        # output moments
  info = FALSE)    # need info?
{
  if (is.null(knots_number) & is.null(knots))
  {
    knots_number = 0
    if (!is.null(t_f) & length(t_f)>0)
      knots_number = length(t_f)
    if (!is.null(t_df) & length(t_df)>0 & knots_number<length(t_df))
      knots_number = length(t_df)
    if (!is.null(t_d2f) & length(t_d2f)>0 & knots_number<length(t_d2f))
      knots_number = length(t_d2f)
    if (!is.null(t_int_a) & length(t_int_a)>0 & knots_number<length(t_int_a))
      knots_number = length(t_int_a)
  }

  if (knots_number<2)
    stop('knots_number or observations should not be less than 2')

  m = knots_number # for short

  # in case knots is not defined
  if (m != length(knots)) # when knots_number defined, but knots not defined
  {
    start_knot = + Inf
    end_knot = - Inf
    if (!is.null(t_f) & length(t_f)>0 )
```

```

    {
        if (start_knot>min(t_f))
            start_knot = min(t_f)
        if (end_knot<max(t_f))
            end_knot = max(t_f)
    }
    if (!is.null(t_df) & length(t_df)>0)
    {
        if (start_knot>min(t_df))
            start_knot = min(t_df)
        if (end_knot<max(t_df))
            end_knot = max(t_df)
    }
    if (!is.null(t_d2f) & length(t_d2f)>0)
    {
        if (start_knot>min(t_d2f))
            start_knot = min(t_d2f)
        if (end_knot<max(t_d2f))
            end_knot = max(t_d2f)
    }
    if (!is.null(t_int_a) & length(t_int_a)>0 & length(t_int_b)>0)
    {
        if (start_knot>min(t_int_a))
            start_knot = min(t_int_a)
        if (end_knot<max(t_int_b))
            end_knot = max(t_int_b)
    }
    knots=seq(start_knot,end_knot,length = m)
}

h = array(0,dim = m - 1) #array of distance between knots
h[1:(m - 1)] = knots[2:m] - knots[1:(m - 1)]

#Matrix Q
Q=matrix(0, nrow = m, ncol = m - 2)
for (i in 1:(m - 2))
{
    Q[i,i] = 1/h[i];
    Q[i + 1,i] = - 1/h[i] - 1/h[i + 1];
    Q[i + 2,i] = 1/h[i + 1]
}

#Matrix R
R = matrix(0, nrow = m - 2, ncol = m - 2)
for (i in 1:(m - 2))
{
    R[i,i] = 1/3*(h[i] + h[i + 1]);
    if (i<m - 2)
    {
        R[i + 1,i] = 1/6*h[i + 1];
        R[i,i + 1] = 1/6*h[i + 1];
    }
}

```

```

#Matrix K calculation
inv_R = solve(R)
t_Q = t(Q)
K = Q %%% inv_R %%% t_Q

# ===== 1. Observation (t_f, values_f) =====

if (!is.null(t_f) & length(t_f)>0)
{
  nf = length(t_f) #number of observation coordinates
  if (length(values_f) != nf)
    stop('length of values_f and t_f must be same')
  if (is.null(weights_f))
    weights_f = rep(1,nf)
  if (length(weights_f) != nf)
    stop('length of weights_f and t_f must be same')
  Wf = diag(weights_f)

  #reorder observations (t_f, values_f) by appear time t_f
  ord = order(t_f,values_f)
  t_f = t_f[ord]
  values_f = values_f[ord]

  #Filling in Vf and Pf matrices
  Vf = matrix(0,nrow = nf, ncol = m)
  Pf = matrix(0,nrow = nf, ncol = m)
  k = 1 # start knot
  for (i in 1:nf)
  {
    while( (knots[k]<=t_f[i]) & (knots[k+1]<t_f[i]) & (k<knots_number)) #find first k, that
knots[k+1]>t_f[i]
      k = k+1
      hk_m = t_f[i] - knots[k]
      hk_p = knots[k+1] - t_f[i]
      Vf[i,k] = hk_p/h[k]
      Vf[i,k + 1] = hk_m/h[k]
      Pf[i,k] = hk_m*hk_p*(h[k] + hk_p)/(6*h[k])
      Pf[i,k + 1] = hk_m*hk_p*(h[k] + hk_m)/(6*h[k])
    }
    Pf = Pf[1:nf,2:(m - 1)] #don't need first and last column

  #Matrix Cf calculation
  Cf = Vf - Pf %%% inv_R %%% t_Q
  t_Cf = t(Cf)
}

# ===== 2. Observation (t_df, values_df) =====

if (!is.null(t_df) & length(t_df)>0)
{

  ndf=length(t_df) #number of observation
  if (length(values_df) != ndf)

```

```

        stop('length of values_df and t_df must be same')
    if (is.null(weights_df))
        weights_df = rep(1,ndf)
    if (length(weights_df)!=ndf)
        stop('length of weights_df and t_df must be same')
    Wdf = diag(weights_df)

    ord = order(t_df,values_df) #reorder observations (t_df, values_df) by appear time t_df
    t_df = t_df[ord]
    values_df = values_df[ord]

    #Filling in Vdf and Pdf matrices
    Vdf = matrix(0, nrow = ndf, ncol = m)
    Pdf = matrix(0, nrow = ndf, ncol = m)
    k = 1 # start knot
    for (i in 1:ndf)
    {
        while( (knots[k]<=t_df[i]) & (knots[k+1]<t_df[i]) & (k<m)) #find first k, that knots[k +
1]>t_df[i]
            k = k + 1
        hk_m = t_df[i] - knots[k]
        hk_p = knots[k + 1] - t_df[i]

        Vdf[i,k] = - 1/h[k]
        Vdf[i,k + 1] = 1/h[k]
        Pdf[i,k] = - h[k]/6+(hk_p)^2/(2*h[k])
        Pdf[i,k + 1] = h[k]/6-(hk_m)^2/(2*h[k])
    }
    Pdf = Pdf[1:ndf,2:(m-1)] #don't need first and last column

    #Matrix Cdf calculation
    Cdf = Vdf - Pdf %*% inv_R %*% t_Q
    t_Cdf = t(Cdf)

}

# ===== 3. Observation (t_d2f, values_d2f) =====

if (! is.null(t_d2f) & length(t_d2f)>0)
{

    nd2f = length(t_d2f) #number of observation
    if (length(values_d2f) != nd2f)
        stop('length of values_d2f and t_d2f must be same')
    if (is.null(weights_d2f))
        weights_d2f = rep(1,nd2f)
    if (length(weights_d2f) != nd2f)
        stop('length of weights_d2f and t_d2f must be same')
    Wd2f = diag(weights_d2f)

    #reorder observations (t_d2f, values_d2f) by appear time t_d2f
    ord = order(t_d2f,values_d2f)
    t_d2f = t_d2f[ord]

```

```

values_d2f = values_d2f[ord]

#Filling in Pd2f matrices
Pd2f=matrix(0, nrow = nd2f,ncol = m)
k = 1 # start knot
for (i in 1:nd2f)
{
  while( (knots[k]<=t_d2f[i]) & (knots[k+1]<t_d2f[i]) & (k<m)) #find first k, that
knots[k+1]>t_d2f[i]
      k = k + 1
      hk_m = t_d2f[i] - knots[k]
      hk_p = knots[k + 1] - t_d2f[i]

      Pd2f[i,k] = - hk_p/h[k]
      Pd2f[i,k+1] = - hk_m/h[k]
}
Pd2f = Pd2f[1:nd2f,2:(m - 1)] #don't need first and last column

#Matrix Cd2f calculation
Cd2f = - Pd2f %%% inv_R %%% t_Q
t_Cd2f = t(Cd2f)

}

# ===== 4. Observation (t_int_a, t_int_b, values_int) =====

if (!is.null(t_int_a) & length(t_int_a)>0)
{
  nint=length(t_int_a) #number of observation
  if (length(t_int_b) != nint)
    stop('length of t_int_b and t_int_a must be same')
  #if (length(values_int) != nint)
    stop('length of values_int and t_int_a must be same')

  if (is.null(weights_int))
    weights_int = rep(1,nint)
  if (length(weights_int) != nint)
    stop('length of weights_int and t_int_a must be same')
  Wint=diag(weights_int)

  #reorder observations (t_int_a, t_int_b, values_int) by appear time t_int_a
  ord = order(t_int_a, t_int_b, values_int)
  t_int_a = t_int_a[ord]
  t_int_b = t_int_b[ord]
  values_int = values_int[ord]

  #Filling in Vint and Pint matrices
  Vint = matrix(0, nrow = nint, ncol = m)
  Pint = matrix(0, nrow = nint, ncol = m)
  k = 1 # start knot
  for (i in 1:nint)
  {
    while( (knots[k]<=t_int_a[i]) & (knots[k + 1]<t_int_a[i]) & (k<m)) #find first k, that

```

```

knots[k + 1] > t_int_a[i]

    k = k + 1

    #finding L, it can be 0
    for (L in 0:(m - k - 1))
        if (t_int_b[i] <= knots[k + L + 1])
            break;

    l = 1;
    hk_m = t_int_a[i] - knots[k]
    hk_p = knots[k + 1] - t_int_a[i]
    hkL_m = t_int_b[i] - knots[k + L]
    hkL_p = knots[k + L + 1] - t_int_b[i]

    Vint[i,k] = (hk_p)^2/h[k]/2
    Pint[i,k] = h[k]^3/24 - (hk_m)^2*(hk_p + h[k])^2/h[k]/24
    while (l <= L)
    {
        Vint[i, k + l] = (h[k + l - 1] + h[k + l])/2
        Pint[i, k + l] = (h[k + l - 1]^3 + h[k + l]^3)/24
        l = l + 1;
    }
    Vint[i, k + 1] = Vint[i, k + 1] - (hk_m)^2/h[k]/2
    Pint[i, k + 1] = Pint[i, k + 1] + (hk_m)^2*((hk_m)^2 - 2*h[k]^2)/h[k]/24
    Vint[i, k + L] = Vint[i, k + L] - (hkL_p)^2/h[k + L]/2
    Pint[i, k + L] = Pint[i, k + L] + (hkL_p)^2*((hkL_p)^2 - 2*h[k + L]^2)/h[k + L]/24
    Vint[i, k + L + 1] = (hkL_m)^2/h[k + L]/2
    Pint[i, k + L + 1] = h[k + L]^3/24 - (hkL_p)^2*(hkL_m + h[k + L])^2/h[k + L]/24

}
Pint = Pint[1:nint, 2:(m - 1)] #don't need first and last column

#Matrix Cint calculation
Cint = Vint - Pint %*% inv_R %*% t_Q
t_Cint = t(Cint)

}

# ===== Calculation =====

# matrix A
A = alpha * K
if (!is.null(t_f) & length(t_f) > 0)
    A = A + t_Cf %*% Wf %*% Cf
if (!is.null(t_df) & length(t_df) > 0)
    A = A + coef_df * t_Cdf %*% Wdf %*% Cdf
if (!is.null(t_d2f) & length(t_d2f) > 0)
    A = A + coef_d2f * t_Cd2f %*% Wd2f %*% Cd2f
if (!is.null(t_int_a) & length(t_int_a) > 0)
    A = A + coef_int * t_Cint %*% Wint %*% Cint

# matrix D
D = matrix(0, nrow = m, ncol = 1)
if (!is.null(t_f) & length(t_f) > 0)

```

```

        D = D + t_Cf %%% Wf %%% values_f
if (!is.null(t_df) & length(t_df)>0)
    D = D + coef_df * t_Cdf %%% Wdf %%% values_df
if (!is.null(t_d2f) & length(t_d2f)>0)
    D = D + coef_d2f * t_Cd2f %%% Wd2f %%% values_d2f
if (!is.null(t_int_a) & length(t_int_a)>0)
    D = D + coef_int * t_Cint %%% Wint %%% values_int

#Calculation of g and gamma
g = solve(A, D)
gamma = inv_R %%% t_Q %%% g #After that spline is completely defined via g and gamma

# ===== Calculating and returning spline values at x coordinates =====

g2 = c(0,gamma,0) #Second derivative on the edges was zero

if (is.null(x))
    x = seq(knots[1],knots[m],by=1)

y = rep(0,length(x))

k = 1; #index of interval
for (j in (1:length(x)))
{
    while (x[j]>knots[k]+h[k] & k<m)
        k = k + 1;
    hk_m = x[j] - knots[k]
    hk_p = knots[k + 1] - x[j]
    y[j] = (hk_m*g[k + 1] + hk_p*g[k])/h[k] - 1/6*hk_m*hk_p*(g2[k + 1]*(1 + hk_m/h[k]) + g2[k]*(1 +
hk_p/h[k]) )
}

if (info)
{
    error_total = 0
    error_f = 0
    error_df = 0
    error_d2f = 0
    error_int = 0
    error_penalty = 0
    fraction_error_f = 0
    fractio_error_df = 0
    fractio_error_d2f = 0
    fractio_error_int = 0
    fractio_penalty = 0
    relative_sqr_error_f = 0
    relative_sqr_error_df = 0
    relative_sqr_error_d2f = 0
    relative_sqr_error_int = 0
    relative_abs_error_f = 0
    relative_abs_error_df = 0
    relative_abs_error_d2f = 0

```

```

relative_abs_error_int = 0

if (!is.null(t_f) & length(t_f)>0)
{
  V = values_f - Cf %>% g
  error_f = t(V) %>% Wf %>% V
  V = abs(V / values_f)
  relative_abs_error_f = (t(V) %>% Wf %>% V) / nf
  V = V^2
  relative_sqr_error_f = sqrt((t(V) %>% Wf %>% V) / nf)
}
if (!is.null(t_df) & length(t_df)>0)
{
  V = values_df - Cdf %>% g
  error_df = t(V) %>% Wdf %>% V
  V = abs(V / values_df)
  relative_abs_error_df = (t(V) %>% Wdf %>% V) / ndf
  V = V^2
  relative_sqr_error_df = sqrt((t(V) %>% Wdf %>% V) / ndf)
}
if (!is.null(t_d2f) & length(t_d2f)>0)
{
  V = values_d2f - Cd2f %>% g
  error_d2f = t(V) %>% Wd2f %>% V
  V = abs(V / values_d2f)
  relative_abs_error_d2f = (t(V) %>% Wd2f %>% V) / nd2f
  V = V^2
  relative_sqr_error_d2f = sqrt((t(V) %>% Wd2f %>% V) / nd2f)
}
if (!is.null(t_int_a) & length(t_int_a)>0)
{
  V = values_int - Cint %>% g
  error_int = t(V) %>% Wint %>% V
  V = abs(V / values_int)
  relative_abs_error_int = (t(V) %>% Wint %>% V) / nint
  V = V^2
  relative_sqr_error_int = sqrt((t(V) %>% Wint %>% V) / nint)
}
error_penalty = t(g) %>% K %>% g

error_total = error_f + coef_df*error_df + coef_d2f*error_d2f + coef_int*error_int + alpha*error_
penalty

fraction_error_f = error_f/error_total
fraction_error_df = coef_df*error_df/error_total
fraction_error_d2f = coef_d2f*error_d2f/error_total
fraction_error_int = coef_int*error_int/error_total
fraction_penalty = alpha*error_penalty/error_total

result = list(  x = x,
                y = y,
                error_total = error_total,
                error_f = error_f,
                error_df = error_df,

```

```

        error_d2f = error_d2f,
        error_int = error_int,
        error_penalty = error_penalty,
        fraction_error_f = fraction_error_f,
        fraction_error_df = fraction_error_df,
        fraction_error_d2f = fraction_error_d2f,
        fraction_error_int = fraction_error_int,
        fraction_penalty = fraction_penalty,
        relative_sqr_error_f = relative_sqr_error_f,
        relative_sqr_error_df = relative_sqr_error_df,
        relative_sqr_error_d2f = relative_sqr_error_d2f,
        relative_sqr_error_int = relative_sqr_error_int,
        relative_abs_error_f = relative_abs_error_f,
        relative_abs_error_df = relative_abs_error_df,
        relative_abs_error_d2f = relative_abs_error_d2f,
        relative_abs_error_int = relative_abs_error_int
    )
}
else
    result = y
return (result) }

```

Приложение 2

Выполнение вычислений в R

Пример 1

Чтение данных из CSV файла (в котором есть все соответствующие столбцы):

```

#===== Data input =====
library(lubridate)
filename = « F:/DIR/Sales.csv»;

#if CSV file was generated by Excel
MyData <- read.csv(file = filename, header = TRUE, sep = «;», stringsAsFactors = FALSE, dec = «,»)
t = as.numeric(dmy(MyData[[1]]))

#if CSV file was generated by R
#MyData <- read.csv(file = filename, header = TRUE, sep = «,», stringsAsFactors = FALSE, dec = «,»)
#t = as.numeric(ymd(MyData[[1]]))

```

Удаление отсутствующих значений:

```

# ===== Remove NA =====
t = t[!is.na(t)]
n = length(t)
Y = MyData[[2]]
Y = Y[1:(n - 1)] # Last value not used
origin = dmy(MyData[[1]][1]) - t[1] #will need time origin for x-axis labes

```

Задаем количество узлов, можно брать в несколько раз больше, чем количество всех наблюдений:

```
m = round(n*3)
```

Вызов функции, результат сохраняется в переменную *r*. Если *Info* = FALSE, то выводиться будут сразу рассчитанные значения *y*.

```
##### Calculating spline #####
r = FunctionalSmoothingSpline(#t_f = t_f,
                              #values_f = y_f,
                              #weights_f = NULL,
                              #t_df = t_df,
                              #values_df = y_df,
                              #weights_df = NULL,
                              #coef_df = 1,
                              #t_d2f = t_d2f,
                              #values_d2f = y_d2f,
                              #weights_d2f = NULL,
                              #coef_d2f = 1,
                              t_int_a = t[1:(n-1)],
                              t_int_b = t[2:n],
                              values_int = Y,
                              weights_int = W,
                              #coef_int = 1,
                              #knots = NULL,
                              knots_number = m,
                              alpha = 10^(4),
                              info = TRUE)

#r
y = r$y
```

Вывод графика:

```
##### Plotting spline with graph of average values #####
x = r$x
#x = seq(t[1], t[n], by = 1) # In case Info = FALSE
x2 = t[1]:t[n] #for graph of average values
y2 = rep(0, length(x2))
for (i in 1:(n - 1))
  for (j in t[i]:t[i + 1])
    y2[j-t[1] + 1] = Y[i]/(t[i + 1] - t[i])

plot(x, y, col = "red", type = "l", lwd = "1", lty = 1, xaxt = "n", ylim = range(c(y,y2)), xlim = range(c(x,x2)))
axis.Date(1, at=seq(min(dmy(MyData[[1]])), max(dmy(MyData[[1]])), by = "months"), format = "%m-%Y")
lines(x2, y2, col = "black", type = "l", lwd = "1", lty = 1)
```

Вывод в файл:

```
##### Data output #####
MyWriteData = data.frame(t = x + origin, Value = y, x2 = x2 + origin, y2 = y2)
s2 = «F:/DIR/f_spline_out.csv»;
write.csv(MyWriteData, file = s2,row.names=FALSE)
```

Пример 2

Чтение данных из CSV файла (в котором есть все соответствующие столбцы):

```
##### Data input #####
```

```

Library (lubridate)
filename = «F:/Dir/DiscrSignals.csv»;

#if CSV file was generated by Excel
MyData <- read.csv(file = filename, header = TRUE, sep = «;», stringsAsFactors = FALSE, dec = «,»)
t_f = as.numeric(dmy(MyData[[1]]))
y_f = MyData[[2]]
t_df = as.numeric(dmy(MyData[[3]]))
y_df = MyData[[4]]
t_d2f = as.numeric(dmy(MyData[[5]]))
y_d2f = MyData[[6]]
t_int_a = as.numeric(dmy(MyData[[7]]))
t_int_b = as.numeric(dmy(MyData[[8]]))
y_int = MyData[[9]]

#if CSV file was generated by R
#MyData <- read.csv(file = filename, header = TRUE, sep = «», stringsAsFactors = FALSE, dec = «.»)
#t_f = as.numeric(ymd(MyData[[1]]))
#t_df = as.numeric(ymd(MyData[[3]]))
#t_d2f = as.numeric(ymd(MyData[[5]]))
#t_int_a = as.numeric(ymd(MyData[[7]]))
#t_int_b = as.numeric(ymd(MyData[[8]]))

```

Удаление отсутствующих значений, идущих в самом конце (в CSV файле длина столбцов была разная, но отсутствующие значения все равно читаются):

```

# ===== Remove NA =====
t_f = t_f[!is.na(t_f)]
nf = length(t_f)
y_f = y_f[1:nf]
t_df = t_df[!is.na(t_df)]
ndf = length(t_df)
y_df = y_df[1:ndf]
t_d2f = t_d2f[!is.na(t_d2f)]
nd2f = length(t_d2f)
y_d2f = y_d2f[1:nd2f]
t_int_a = t_int_a[!is.na(t_int_a)]
nint = length(t_int_a)
t_int_b = t_int_b[!is.na(t_int_b)]
y_int = y_int[1:nint]

#will need time origin for x-axis labes
origin = dmy(MyData[[1]][1]) - t_f[1]
#origin = ymd(MyData[[1]][1]) - t_f[1]

```

Задаем количество узлов, можно брать в несколько раз больше, чем количество всех наблюдений:

```
m = round(3*(nf + ndf + nd2f + nint))
```

Вызов функции, результат сохраняется в переменную *r*. Если *Info* = FALSE, то выводиться будут сразу рассчитанные значения *y*:

```
##### Calculating spline #####
```

```
r = FunctionalSmoothingSpline(t_f = t_f,
                              values_f = y_f,
                              t_df = t_df,
                              values_df = y_df,
                              t_d2f = t_d2f,
                              values_d2f = y_d2f,
                              t_int_a = t_int_a,
                              t_int_b = t_int_b,
                              values_int = y_int,
                              knots_number = m,
                              alpha = 10^(0),
                              info = TRUE)
```

```
y = r$y
```

Вывод графика:

```
##### Plotting spline #####
```

```
x = r$x
#x = seq(min(t_f, t_df, t_d2f, t_int_a), max(t_f, t_df, t_d2f, t_int_b), by = 1) # In case Info = FALSE
xrange = range(x)
yrange = range(y)
plot(x, y, col = "red", type = "l", lwd = "1", lty = 1, xaxt = "n", ylim = yrange, xlim = xrange)
axis(1, at = c(t_f, t_df, t_d2f, t_int_a, t_int_b), labels = as.Date(c(t_f, t_df, t_d2f, t_int_a, t_int_b), origin=origin))
#lines(x,y2,col = "black", type = "l", lwd="2", lty = 1) # In case you want add some precalculated original
function y2
```

Вывод в файл:

```
##### Data output #####
```

```
#write to the csv file
MyWriteData = data.frame(x + origin, y)
s2 = «F:/DIR/f_spline_out.csv»
write.csv(MyWriteData, file = s2, row.names = FALSE)
```

ИНФОРМАЦИЯ ОБ АВТОРЕ / ABOUT THE AUTHOR



Юрий Александрович Кораблев — кандидат экономических наук, доцент, доцент кафедры системного анализа в экономике, Финансовый университет, Москва, Россия
Yurii A. Korablev — Cand. Sci. (Econ.), Assoc. Prof., Department of System Analysis in Economics, Financial University, Moscow, Russia
<http://orcid.org/0000-0001-5752-4866>
yura-korablyov@yandex.ru

Конфликт интересов: автор заявляет об отсутствии конфликта интересов.

Conflicts of Interest Statement: The author has no conflicts of interest to declare.

Статья поступила в редакцию 21.10.2021; после рецензирования 08.11.2021; принята к публикации 25.02.2022.

Автор прочитал и одобрил окончательный вариант рукописи.

The article was submitted on 21.10.2021; revised on 08.11.2021 and accepted for publication on 25.02.2022.

The author read and approved the final version of the manuscript.