

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Фізико-математичний факультет

Кафедра математичного аналізу та теорії ймовірностей

«На правах рукопису»
УДК 519.21

«До захисту допущено»
Завідувач кафедри
_____ О.І. Клесов
«08» червня 2022 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-науковою програмою «Страхова та фінансова математика»

зі спеціальності 111 «Математика»

**на тему: «Дослідження кореляційних зв'язків ринкових показників під час світової
фінансової кризи»**

Виконав:

студент II курсу магістратури, групи ОМ-01мн

Мучак Максим Олександрович _____

Керівник:

кандидат фізико-математичних наук, доцент

Буценко Юрій Павлович _____

Рецензент:

кандидат фізико-математичних наук, доцент

Каніовська Ірина Юріївна _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.

Студент _____

Київ - 2022 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Фізико-математичний факультет

Кафедра математичного аналізу та теорії ймовірностей

Рівень вищої освіти – другий (магістерський)

Спеціальність – 111 «Математика»

Освітньо-наукова програма «Страхова та фінансова математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

О.І. Клесов

« 03 » лютого 2022 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Мучаку Максиму Олександровичу

1. Тема дисертації: «Дослідження кореляційних зв'язків ринкових показників під час світової фінансової кризи», науковий керівник дисертації: к.ф.-м.н., доц. Буценко Юрій Павлович затверджені наказом по університету від « 25 » травня 2022 р. № НС/372/2022.
2. Термін подання студентом дисертації: 07 червня 2022 р.
3. Об'єкт дослідження: стохастичні процеси в економічних системах.
4. Предмет дослідження: кореляційний зв'язок між траєкторіями вартості акцій підприємств, з незалежними структурами власності, ринками та місцезнаходженням.
5. Перелік завдань, які потрібно розробити:
 - 1) Ознайомлення з літературою з математичного моделювання ринкових показників.
 - 2) Освоєння практичних навичок дослідження траєкторій ринкових показників.
 - 3) Отримання статистичних оцінок для зв'язків між траєкторіями акцій різних підприємств.
 - 4) Розробка програмного забезпечення для прогнозування сукупностей траєкторій сильнокорельованих акцій.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: приклади роботи комп'ютерної програми, що моделює розв'язки стохастичних диференціальних рівнянь, а також графічні зображення відповідності побудованих прогнозів реальному перебігу відповідних процесів, порівняння використання пропонованої методики до збільшення пенсійного капіталу з традиційними методами заощадження.

7. Дата видачі завдання 2 лютого 2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Ознайомлення з літературою	23.02.2022 – 18.03.2022	Виконано
2	На основі отриманих кореляційних залежностей будуюмо кластери сильно/слабко залежних компаній, на основі машинного навчання отримуємо відповідні результати	19.03.2022 – 06.04.2022	Виконано
3	Будуємо оптимізаційну модель на основі стратегії мішаного пакету акцій ф'ючерса, складеного з акцій сильнозалежних компаній	07.04.2022 – 07.05.2022	Виконано
4	Моделювання стохастичних процесів та перевірка попередніх теоретичних розрахунків	08.05.2022 – 18.05.2022	Виконано
5	Оформлення роботи	19.05.2022 – 05.06.2022	Виконано

Студент

М.О. Мучак

Науковий керівник

Ю.П. Буценко

РЕФЕРАТ

Магістерська робота містить 69 сторінок, 19 першоджерел, 18 ілюстрацій, 16 слайдів для проєктора.

Структура роботи включає вступ, три основні розділи, висновок та додаток з переліком програмних кодів.

Метою даної роботи є впровадження прогнозування, пошук закономірностей у сучасній фінансовій математиці за допомогою програмних засобів на основі сучасної стохастичної фінансової теорії.

Об'єктом дослідження було обрано клас розв'язків стохастичних диференціальних рівнянь, оскільки їх поведінку можна теоретично визначати моментами перших порядків, що дає змогу впоратись із поставленими задачами оцінки параметрів дифузійних моделей за наявними даними.

Предметом дослідження стали оцінки кореляційного зв'язку між траєкторіями фінансових показників та їх моделювання і прогнозування.

У першому розділі ми розглянули основні теоретичні поняття теорії стохастичних диференціальних рівнянь, економічної теорії та пропонуємо теореми, на яких базуються подальші докази та робочі результати. Основним першоджерелом цього розділу є праці [1]. [2] А. Н. Ширяєва та Б. Оксендаля, які виклали сучасний стан математичної теорії фінансів і теорії стохастичних рівнянь.

У другій частині ми розглянули важливі типи моделей, необхідних для аналізу реальних моделей. Розглянуто методи Кесслера, Ейлера, Ейлера-Маруями, Озакі, Шоджі-Озакі. У цьому розділі ми спиралися на роботу [5].

Третя частина присвячена побудові передбачень на реальних даних з використанням попередньо наведених методів та моделей. Розглянуто пошук закономірностей та побудова портфелю ф'ючерсів на основі реальних даних, використовуючи мови програмування.

ABSTRACT

The master's thesis contains 69 pages, 19 primary sources, 18 illustrations, 16 slides for the projector.

The structure of the work includes an introduction, three main sections, a conclusion and an appendix with a list of program codes.

The purpose of this work is the introduction of forecasting, the search for patterns in modern financial mathematics using software based on modern stochastic financial theory.

The object of the study was the class of solutions of stochastic differential equations, since their behavior can be theoretically determined by first-order moments, which makes it possible to cope with the tasks of estimating the parameters of diffusion models based on available data.

The subject of the study was the assessment of the correlation between the trajectories of financial indicators and their modeling and forecasting.

In the first section, we consider the basic theoretical concepts of the theory of stochastic differential equations, economic theory and propose theorems on which further evidence and working results are based. The main source of this section is the work [1]. [2] AN Shiryaev and B. Oksendal, who outlined the current state of mathematical theory of finance and the theory of stochastic equations.

In the second part, we considered the important types of models needed to analyze real models. The methods of Kessler, Euler, Euler-Maruyami, Ozaki, Shoji-Ozaki are considered. In this section we relied on the work [5].

The third part is devoted to building predictions on real data using the previously described methods and models. The search for patterns and construction of a futures portfolio based on real data using programming languages is considered.

ЗМІСТ

<i>Магістерська дисертація</i>	1
<i>РЕФЕРАТ</i>	4
<i>ABSTRACT</i>	5
<i>ЗМІСТ</i>	6
<i>ВСТУП</i>	7
1 ТЕОРЕТИЧНА ЧАСТИНА	8
1.1 Вінерівський процес	8
1.2 Інтеграл Іто	9
1.3 Стохастичні диференціальні рівняння	10
1.4 Коваріація	12
1.5 Відомості з економічної теорії	13
2. МЕТОДИ ЕЙЛЕРА-МАРУЯМИ, ЕЙЛЕРА, КЕССЛЕРА, ОЗАКІ, ШОДЖІ-ОЗАКІ	14
2.1 Метод Ейлера-Маруями	14
2.2 Метод Ейлера	16
2.3 Метод Кесслера	18
2.4 Метод Озакі	19
2.5 Метод Шоджі-Озакі	20
3. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ	22
3.1 Моделювання процесу на прикладі висококорельованих компаній в кризовий період	22
3.2 Приклади застосування виявлених стохастичних закономірностей та методів прогнозування	32
3.3 Моделювання портфеля ф'ючерсів	33
ВИСНОВКИ	37
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	38
ДОДАТОК	40

ВСТУП

Сучасна фінансова математика широко використовує інструменти теорії стохастичних диференціальних рівнянь. Поява інтеграла Іто дозволяє створити нову платформу для вивчення та прогнозування випадкових процесів як у галузі фінансів, дослідження фондових ринків, так і в інших науках: фізиці, біології, соціології.

У нашій роботі ми розглянемо практичне застосування на реальних даних стохастичних процесів, які дозволяють нам будувати прогнози. Зокрема, значну увагу буде приділено методам Кеслера, Ейлера які будуть широко використовуватися для побудови прогнозів. Також, обговорюватиметься питання застосування виявлених стохастичних закономірностей та побудова портфеля ф'ючерсів.

У теоретичному дослідженні ми будемо спиратися на фінансову математику Ширяєва А.Н., теорію стохастичних диференціальних рівнянь Б. Оксендала, роботи Guidoum AC, Boukhetala K.

Наша основна мета – навчитися найбільш ефективно прогнозувати та досліджувати закономірності, використовуючи арсенал математичних методів та обчислювальні потужності комп'ютера і серверів.

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Вінерівський процес

Основною концепцією для побудови теорії стохастичних диференціальних рівнянь є броунівський рух або процес Вінера. Це пов'язано головним чином з тим, що за допомогою приростів цього процесу було успішно побудовано інтеграл нового типу.

Означення 1. Процес $w_t, t \geq 0$ називається вінерівським, якщо:

1) w_t є гаусовим, тобто

$\forall n \in \mathbb{N} \forall t_1, t_2, \dots, t_n : (w_{t_1}, w_{t_2}, \dots, w_{t_n})^T$ — гаусовий вектор.

2) $E w_t = 0, t \geq 0$

3) $cov(w_t, w_s) = \min(t, s), t, s \geq 0$

Основні властивості:

1) $w_t \sim N(0, t)$

w_t має неперервну за t модифікацію, тобто існує процес $\widetilde{w}_t$ — вінерівський і неперервний за часом такий, що:

$$P(w_t = \widetilde{w}_t) = 1$$

2) Незалежність приростів. $\forall n \in \mathbb{N} \forall 0 \leq t_1 \leq t_2 \leq \dots \leq t_n$ незалежними у сукупності є випадкові величини

$$w_{t_1}, w_{t_2} - w_{t_1}, \dots, w_{t_n} - w_{t_{n-1}}$$

3) Однорідність приростів

$$\forall s, t \geq 0: w_{t+s} - w_t \sim w_s$$

4) Вінерівський процес недиференційовний м.н., тобто

$$\forall t \geq 0 P(\exists w'_t) = 0$$

1.2 Інтеграл Іто

Нехай F_t потік σ -алгебр:

$$\forall t_1 \leq t_2 : F_{t_1} \subset F_{t_2}$$

Означення 2. Вінерівський процес w_t , $t \geq 0$ називається узгодженим з потоком F_t , якщо:

- 1) $\forall t \geq 0 : w_t$ — F_t -вимірнав.в.
- 2) $\forall t \geq 0, s \geq 0 : w_{t+s} - w_t$ незалежні від F_t

Спочатку інтеграл Іто будується для простих процесів вигляду:

$$\xi_t = \sum_{k=0}^{n-1} \alpha_k I_t \in (t_k + t_{k+1}],$$

де випадкові величини α_k обмежені та вимірні відносно F_{t_k} ,

$0 \leq t_1 \leq t_2 \leq \dots \leq t_n = T$. Клас таких процесів будемо позначати $\mathcal{L}_0$.

Означення 3. Нехай $\xi_t \in \mathcal{L}_0$, w_t — узгоджений вінерівський процес.

Інтеграл Іто від простого процесу будемо називати випадковою

$$\int_0^T \xi_t dw_t := \sum_{k=0}^{n-1} \alpha_k (w_{t_{k+1}} - w_{t_k})$$

Основні властивості:

Якщо $\xi_t, \eta_t \in \mathcal{L}_0$; $a, b \in \mathbb{R}$, то

- 1) $\int_0^T (a\xi_t + b\eta_t) dw_t = a \int_0^T \xi_t dw_t + b \int_0^T \eta_t dw_t$
- 2) $E \int_0^T \xi_t dw_t = 0$
- 3) $E (\int_0^T \xi_t dw_t)^2 = E \int_0^T \xi_t^2 dw_t$
- 4) $E (\int_0^T \xi_t dw_t \int_0^T \eta_t dw_t) = E \int_0^T \xi_t \eta_t dw_t = \int_0^T E(\xi_t \eta_t) dw_t$

З останньої властивості випливає, що інтеграл Іто зберігає норму. Якщо процес можна наблизити простим процесом, то існує можливість здійснити граничний перехід в просторі L_2 . Отже, інтеграл Іто продовжується на замиканні $\mathcal{L}_0$.

1.3 Стохастичні диференціальні рівняння

Означення 4. Процес $\xi_t = \xi(\omega, t)$ називається прогресивно-вимірним, якщо $\forall t \in [0, T] : \xi(\cdot, \cdot) \in F_t \times B([0, t])$ -вимірним відображенням.

Означення 5. Стохастичним диференціалом для прогресивно-вимірного процесу $\xi_t, t \in [0, T]$ будемо називати $d\xi_t = a(t)dt + b(t)dw_t$, якщо:

1) a, b – прогресивно-вимірні

2)

$$\int_0^T |a(t)|dt < \infty, \int_0^T b^2(t)dt < \infty \text{ м.н.}$$

3)

$$\xi_t = \xi_0 + \int_0^t a(s)ds + \int_0^t b(s)dw_s \text{ м.н.}$$

Наступна теорема, вперше доведена К. Іто є центральною у стохастичному диференціальному численні. Її доведення можна знайти у працях [2][3][4]

Теорема 1. Формула Іто

Нехай $f = f(t, x) \in C^{1,2}([0, T] \times \mathbb{R})$

Також припустимо, що

$$d\xi_t = a(t)dt + b(t)dw_t$$

Тоді

$$df(t, \xi_t) = f'_t(t, \xi_t)dt + f'_x(t, \xi_t)a(t)dt + f'_x(t, \xi_t)b(t)dw_t + \frac{1}{2}f''_{xx}(t, \xi_t)b^2(t)dt$$

Формула Іто дозволяє звести одні рівняння до інших рівнянь і ми знаємо розв'язки цих рівнянь.

Означення 6. Випадковий процес $\xi_t, t \in [0, T]$ є розв'язком стохастичного диференціального рівняння

$$d\xi_t = a(t, \xi_t)dt + b(t, \xi_t)dw_t, t \in [0, T]$$

з початковою умовою $\xi(0) = \xi_0$, якщо

1) $\forall t \in [0, T] : \xi_t - F_t$ -вимірний

2) ξ_t – неперервний по t м.н.

3)

$$\xi_t = \xi_0 + \int_0^t a(s, \xi_s) ds + \int_0^t b(s, \xi_s) dw_s \quad \forall t \in [0, T] \text{ м.н.}$$

Наведемо теорему існування і єдиності розв'язку стохастичного диференціального рівняння. Її можна знайти, наприклад, у роботі [2].

Теорема 2. *Припустимо, що*

1) $E(\xi_0)^2 < \infty$

2) a, b задовольняють умову Лівшиця по x

3) a, b задовольняють умову лінійного росту по x

Тоді існує розв'язок СДР такий, що

1) $E \max_{t \in [0, T]} \xi_t^2 < \infty$

2) Для будь-якого розв'язку $\eta_t, t \in [0, T]$ такого, що $E \max_{t \in [0, T]} \eta_t^2 < \infty$

виконується:

$$P(\xi_t = \eta_t, t \in [0, T]) = 1$$

1.4 Коваріація

Для двох спільно розподілених випадкових величин X і Y з кінцевими моментами другого порядку, коваріація визначається як середнє добутку їх відхилень від відповідних очікуваних значень [7]:

$$\text{cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

Де $E(X)$ - очікуване значення X . За аналогією з дисперсією, коваріацію також іноді називають $\sigma(X, Y)$. Використовуючи очікувану лінійну властивість, це можна спростити:

$$\begin{aligned}\text{cov}(X, Y) &= E[(X - E[X])(Y - E[Y])] = E[XY - XE[Y] - E[X]Y + E[X]E[Y]] \\ &= E[XY] - E[X]E[Y] - E[X]E[Y] + E[X]E[Y] = E[XY] - E[X]E[Y]\end{aligned}$$

Для Дискретних випадкових величин:

Дійсна пара випадкових змінних (X, Y) може приймати значення (x_i, y_i) для $i = 1, \dots, n$ де ймовірність: $p_i = \frac{1}{n}$, то коваріацію можна записати як $E[X]$ і $E[Y]$:

$$\text{cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - E(X))(y_i - E(Y)).$$

Основні властивості:

$$1) \quad \text{cov}(X, X) = \text{var}(X) = \sigma^2(X)$$

$$2) \quad \text{cov}(X, a) = 0$$

$$3) \quad \text{cov}(X, Y) = \text{cov}(Y, X)$$

$$4) \quad \text{cov}(aX, bY) = ab \text{ cov}(X, Y)$$

$$5) \quad \text{cov}(X + a, Y + b) = \text{cov}(X, Y)$$

6) Якщо X, Y незалежні випадкові величини, то коваріація рівна 0 із – за $E[XY] = E[X]E[Y]$

1.5 Відомості з економічної теорії

Для довідки зазначимо, що таке фондовий ринок акцій та наведемо основні терміни, які використовуватимуться у майбутньому:

Фондовий ринок – це біржова торгівля корпоративними правами. Акціонерні товариства можуть залучати кошти шляхом додаткової емісії для розвитку бізнесу. Молоді компанії отримують стартовий капітал за рахунок продажу частини статутного капіталу. Акціонери компанії очікують, що їхні акції з часом подорожчатимуть, а частина прибутку виплачуватиметься у вигляді дивідендів.

High price - найвища ціна акцій за торгову сесію,

Low price - найнижча ціна акцій за торгову сесію,

Close price - ціна акцій на кінець торгової сесії,

Open price - ціна акцій на початку торгової сесії,

Adjusted close - ціна закриття, скоригована для розподілу дивідендів.

2. МЕТОДИ ЕЙЛЕРА-МАРУЯМИ, ЕЙЛЕРА, КЕССЛЕРА, ОЗАКІ, ШОДЖІ-ОЗАКІ

Для прогнозування розв'язків стохастичного диференціального рівняння необхідно побудувати методи оцінювання їх коефіцієнтів, вказані методи чисельного розв'язку таких рівнянь є найпоширенішими. Розглянемо їх детальніше.

2.1 Метод Ейлера-Маруями

В обчисленні Іто метод Ейлера-Маруями [8] є методом наближеного чисельного розв'язку стохастичного диференціального рівняння (СДР). Це розширення методу Ейлера для звичайних диференціальних рівнянь на СДР.

Розглянемо стохастичне диференціальне рівняння

$$dX_t = a(X_t, t)dt + b(X_t, t)dW_t,$$

з початковим станом $X_0 = x_0$, де W_t представляє процес Вінера. Припустимо, ми хочемо розв'язати це СДР на інтервалі $[0, T]$. Тоді, наближення Ейлера-Маруями до рішення X є ланцюгом Маркова Y , визначеним таким чином:

1) Розбиття інтервалу $[0, T]$ на N рівних під інтервалів однакової довжини $\Delta t > 0$:

$$0 = \tau_0 < \tau_1 < \dots < \tau_N = T \text{ і } \Delta t = \frac{T}{N};$$

2) Встановимо $Y_0 = x_0$

3) Визначимо Y_n при $0 \leq n \leq N - 1$

$$Y_{n+1} = Y_n + a(Y_n, \tau_n)\Delta t + b(Y_n, \tau_n)\Delta W_n,$$

$$\text{де } \Delta W_n = W_{\tau_{n+1}} - W_{\tau_n}$$

Розглянемо приклад зображення Процеса Орнштейна–Уленбека [9].

Процес Орнштейна-Уленбека є стаціонарним процесом Гаусса-Маркова

Процес має рівняння

$$dY_t = \theta \cdot (\mu - Y_t)dt + \sigma dW_t$$

Виконання зображення процесу відбувається за допомогою мови програмування Python, з використанням бібліотек для масивів numpy та для зображення matplotlib.pyplot. Код з поясненнями можна знайти в розділі Додатків.

Зображення метода:

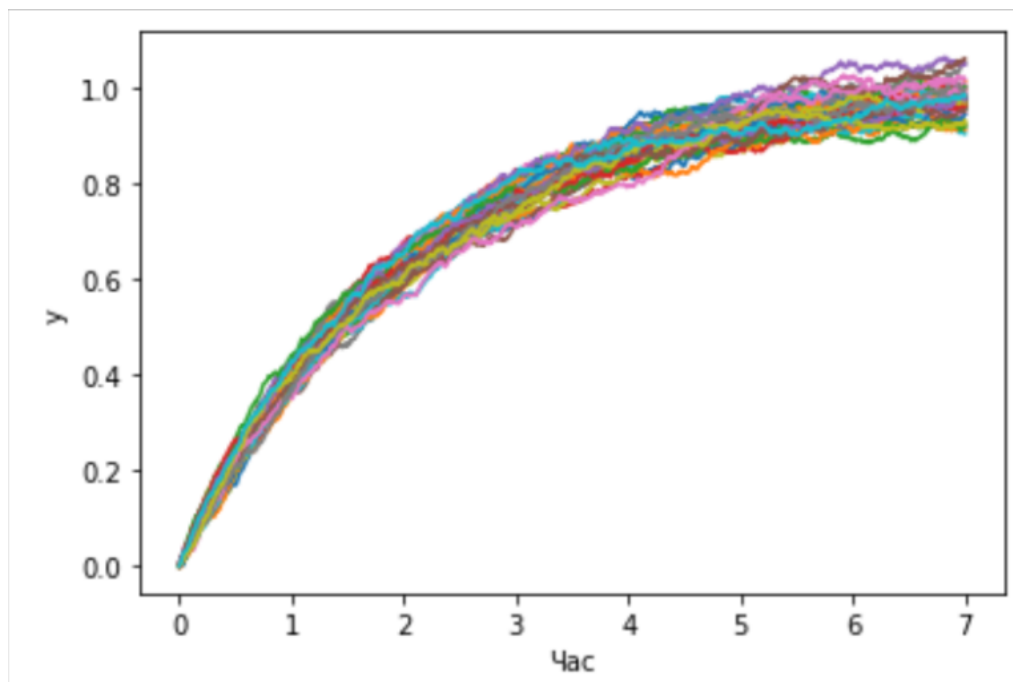


Рис. 1 метод Ейлера-Маруями до процесу Орнштейна-Уленбека

Для побудови цього зображення було задано $\theta = 0.5$, $\mu = 1$, $\sigma = 0.03$ і 50 симуляцій.

2.2 Метод Ейлера

Розглянемо процес розв'язання загального стохастичного диференціального рівняння:

$$dX_t = f(X_t, \bar{\theta})dt + g(X_t, \bar{\theta})dW_t, t \geq 0, X_0 = x_0,$$

де $\bar{\theta}$ є вектор параметрів, що підлягає оцінюванню.

Схема Ейлера [5] перетворює неперервні диференціальні рівняння в дискретне різницеве рівняння, тобто таке, яке пов'язує значення невідомої функції в будь-якій точці з її значенням в одній або кількох точках від заданого інтервалу.

Маємо:

$$X_{t+\Delta t} - X_t = f(X_t, \bar{\theta})\Delta t + g(X_t, \bar{\theta})(W_{t+\Delta t} - W_t)$$

Тоді прирости $X_{t+\Delta t} - X_t$ є незалежними гаусовими випадковими величинами із середніми значеннями: $EX_t = f(X_t, \bar{\theta})\Delta t$ і дисперсією: $DX_t = g^2(X_t, \bar{\theta})\Delta t$. Перехідну щільність процесу можна записати так:

$$p_{\theta}(t, y | x) = \frac{1}{\sqrt{2\pi t g^2(x, \theta)}} \exp \left(-\frac{(y - x - f(x, \theta)t)^2}{2t g^2(x, \theta)} \right)$$

Тоді, логарифмічна функція правдоподібності має вигляд:

$$h_n(\theta | X_1, X_2, \dots, X_n) = -\frac{1}{2} \left(\sum_{i=1}^n \frac{(X_i - X_{i-1} - f(X_{i-1}, \theta)\Delta)^2}{\sigma^2 \Delta t} + n \log(2\pi \sigma^2 \Delta t) \right)$$

Розглянемо приклад:

Модель Чана-Каролі-Лонгстаффа-Сандерса [10]:

$$dX_t = (\theta_1 + \theta_2 X_t)dt + \theta_3 X_t^{\theta_4} dW_t, X_0 = 2$$

Ми робимо припущення про структуру коефіцієнтів СДР та будуємо оцінки параметрів, що входять до цих коефіцієнтів, що матимуть властивість квазімаксимальної правдоподібності [11][12] параметрів зсуву та дифузії.

Для цього скористаємося функцією *fitsde* [16] в мові програмування R, вважаючи що зсув моделі є $\theta_1 + \theta_2 X_t$, дифузії $\theta_3 X_t^{\theta_4}$. В оптимізатор ми виберемо початкові значення

$$\theta_1 = \theta_2 = \theta_3 = \theta_4 = 1, X_0 = 2.$$

```
Method: Euler
Call:
fitsde(data = mydata, drift = fx, diffusion = gx, start = list(theta1 = 1,
  theta2 = 1, theta3 = 1, theta4 = 1), pmle = "euler")

Coefficients:
      Estimate Std. Error
theta1 1.1086528 1.46115207
theta2 1.8618791 0.19199187
theta3 0.4644793 0.01013611
theta4 0.3291023 0.01084505

-2 log L: -66532.06
```

Рис. 2 метод Ейлера для моделі Чана-Каролі-Лонгстаффа-Сандерса

Результат $\theta_1=1.1$, $\theta_2=1.86$, $\theta_3=0.46$, $\theta_4=0.32$ з можливим стандартним відхиленням $\sigma_1 = 1.4611$, $\sigma_2 = 0.1919$, $\sigma_3 = 0.01$ і $\sigma_4 = 0.01$. Слід зазначити, що -66532,06 це значення критерію Шварца.

Означення 7

Критерій Шварца [13] (Баєсівський інформаційний критерій) визначають за формулою:

$$\text{BIC} = \ln(n)k - 2\ln(\hat{L}), \text{ де}$$

1) $\hat{L}$ - максимальне значення функції ймовірності моделі M, тобто $\hat{L} = p(x | \hat{\theta}, M)$, де $\hat{\theta}$ – значення параметра, що максимізує функцію правдоподібності,

2) x – дані спостереження,

3) n – число спостережень,

4) k – число параметрів, які потрібно оцінити.

Вказане значення критерія підлягає мінімізації.

2.3 Метод Кесслера

Кесслер [5] запропонував використовувати розвинення другого порядку для апроксимації середнього і дисперсії умовних гауссових густин, враховуючи СДР СДР

$$dX_t = f(X_t, \bar{\theta})dt + g(X_t, \bar{\theta})dW_t,$$

де $\bar{\theta}$ є вектор параметрів, що підлягає оцінюванню.

Щільність переходу Кесслер:

$$X_{t+\Delta t} | X_t=x \sim \mathcal{N}(E_x, D_x), \text{ де:}$$

$$E_x = x + f(t, x)\Delta t + \left(f(t, x) \partial_x f(t, x) + \frac{1}{2} g^2(t, x) \partial_{xx} g(t, x) \right) \frac{(\Delta t)^2}{2}$$

$$D_x = x^2 + (2f(t, x)x + g^2(t, x))\Delta t + (2f(t, x)(\partial_x f(t, x)x + f(t, x) + g(t, x) \partial_x g(t, x)) + g^2(t, x)(\partial_{xx} f(t, x)x + 2 \partial_x f(t, x) + \partial_x g^2(t, x) + g(t, x) \partial_{xx} g(t, x))) \frac{(\Delta t)^2}{2} - E_x^2.$$

Розглянемо приклад:

Модель Халла-Уайта [14]:

$$dX_t = a(t)(b(t) - X_t)dt + \sigma(t)dW_t, X_0 = 2$$

Ми робимо припущення про структуру коефіцієнтів СДР та будуємо оцінки параметрів, що входять до цих коефіцієнтів, що матимуть властивість квазімаксимальної правдоподібності [11][12] параметрів зсуву та дифузії.

Для цього скористаємося функцією *fitsde* [16] в мові програмування R, вважаючи що зсув моделі є $\theta_1 t (\theta_2 \sqrt{t} - x)$, дифузії $\theta_3 t$. В оптимізатор ми виберемо початкові значення

$$\theta_1 = \theta_2 = \theta_3 = 1, X_0 = 2.$$

```

Pseudo maximum likelihood estimation

Method: Kessler
Call:
fitsde(data = mydata, drift = fx, diffusion = gx, start = list(theta1 = 1,
  theta2 = 1, theta3 = 1), pmle = "kessler")

Coefficients:
      Estimate Std. Error
theta1  3.175948  0.388199697
theta2  1.029409  0.169447610
theta3  0.303182  0.006783359

-2 log L: -8443.074

```

Рис. 3 метод Кесслера для моделі Халла-Уайта

Результат $\theta_1=3$, $\theta_2=1$ і $\theta_3=0.3$ з можливим стандартним відхиленням $\sigma_1 = 0.388$, $\sigma_2 = 0.17$, $\sigma_3 = 0.01$.

2.4 Метод Озакі

Метод Озакі [5] підходить для однорідних стохастичних диференціальних рівнянь. Розглянемо стохастичне диференціальне рівняння такого вигляду:

$$dX_t = f(X_t, \bar{\theta})dt + \sigma dW_t, t \geq 0, X_0 = x_0,$$

де $\bar{\theta}$ є вектор параметрів, що підлягає оцінюванню, σ має бути константою.

Щільність переходу для методу Озакі є гауссовою, маємо, що: $X_{t+\Delta t} | X_t=x \sim \mathcal{N}(E_x, D_x)$, де:

$$E_x = x + \frac{f(x)}{\partial_x f(x)} (e^{\partial_x f(x) \Delta t} - 1), \text{ і } D_x = \sigma^2 \frac{e^{2K_x \Delta t} - 1}{2K_x},$$

Завжди можна трансформувати процес X_t з постійним коефіцієнтом дифузії за допомогою перетворення Ламперті $Z_t = \phi(X_t, t) = \int \frac{1}{\sigma(x,t)} dx$.

Розглянемо приклад:

Модель Васичека [15]:

$$dX_t = \theta_1(\theta_2 - X_t)dt + \theta_3 dW_t, X_0 = 5$$

Ми робимо припущення про структуру коефіцієнтів СДР та будуємо оцінки параметрів, що входять до цих коефіцієнтів, що матимуть властивість квазімаксимальної правдоподібності [11][12] параметрів зсуву та дифузії.

Для цього скористаємося функцією *fitsde* [16] в мові програмування R, вважаючи що зсув моделі є $\theta_1(\theta_2 - x)$, дифузії θ_3 . В оптимізатор ми виберемо початкові значення

$$\theta_1 = \theta_2 = \theta_3 = 1, X_0 = 5.$$

```
Method:  ozaki
Call:
fitsde(data = HWV, drift = fx, diffusion = gx, start = list(theta1 = 1,
  theta2 = 1, theta3 = 1), pmle = "ozaki")

Coefficients:
      Estimate Std. Error
theta1  3.377692  0.39850188
theta2  1.995110  0.04913982
theta3  0.504653  0.01128601

-2 log L: -3135.625
```

Рис. 4 метод Озакі для моделі Васичека

Результат $\theta_1=3.37$, $\theta_2=1.99$ і $\theta_3=0.50$ з можливим стандартним відхиленням $\sigma_1 = 0.398$, $\sigma_2 = 0.04$, $\sigma_3 = 0.01$.

2.5 Метод Шоджі-Озакі

Розширенням до методу Озакі, коли зсув і дифузія залежить від змінної часу t є метод Шоджі-Озакі [5]. Розглянемо стохастичне диференціальне рівняння:

$$dX_t = f(t, X_t, \bar{\theta})dt + g(X_t, \bar{\theta})dW_t, t > 0, X_0 = x_0,$$

де $\bar{\theta}$ є вектор параметрів, що підлягає оцінюванню.

Щільність переходу методу Шоджі-Озакі є гауссовою. Маємо: $X_{t+\Delta t} | X_t = x \sim \mathcal{N}(A_{(t,x)}x, B_{(t,x)}^2)$, де:

$$A_{(t,x)} = 1 + \frac{f(t, x)}{xL_t}(e^{L_t\Delta t} - 1) + \frac{M_t}{xL_t^2}(e^{L_t\Delta t} - 1 - L_t\Delta t)$$

$$B_{(t,x)} = g(x) \sqrt{\frac{e^{2L_t\Delta t} - 1}{2L_t}}$$

$$L_t = \partial_x f(t, x) \text{ і } M_t = \frac{g^2(x)}{2} \partial_{xx} f(t, x) + \partial_t f(t, x)$$

Розглянемо приклад:

Розглянемо наступну модель:

$$dX_t = a(t)X_t dt + \theta_2 X_t dW_t, \quad X_0 = 10$$

Ми робимо припущення про структуру коефіцієнтів СДР та будуємо оцінки параметрів, що входять до цих коефіцієнтів, що матимуть властивість квазімаксимальної правдоподібності [11][12] параметрів зсуву та дифузії.

Для цього скористаємося функцією *fitsde* [16] в мові програмування R, вважаючи що зсув моделі є $\theta_1 x t$, дифузії $\theta_2 x$. В оптимізатор ми виберемо початкові значення

$$\theta_1 = \theta_2 = 1, X_0 = 10.$$

```
Pseudo maximum likelihood estimation

Method:  Shoji
Call:
fitsde(data = mydata, drift = fx, diffusion = gx, start = list(theta1 = 1,
  theta2 = 1), pmle = "shoji", lower = c(-3, 0), upper = c(-1,
  1))

Coefficients:
              Estimate Std. Error
theta1 -2.0188558  0.350251073
theta2  0.2020946  0.004520879

-2 log L: -3444.291
```

Рис. 5 метод Шоджі-Озакі

Результат $\theta_1 = -2.0188$, $\theta_2 = 0.2020$ з можливим стандартним відхиленням $\sigma_1 = 0.3502$, $\sigma_2 = 0.004$.

3. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ

З розвитком комп'ютерних технологій стало можливо використовувати комп'ютерне моделювання для вивчення випадкових процесів та їх прогнозування, виконувати оптимізацію алгоритма. Тепер для цих дій лише потрібно використати готову бібліотеку для отримання бажаного нами результату.

3.1 Моделювання процесу на прикладі висококорельованих компаній в кризовий період

Розглянемо період світової кризи в період 2008-2009 роках. Знайдемо сильнокорельовані компанії з індекса S&P 500. Виберемо пару таких, що не мають нічого спільного між собою, тобто таких, що займаються різною діяльністю, мають різне географічне положення.

Побудуємо для них прогнозування під час даного часового періоду 2008-2009. Виберемо дані в дискретній формі, так як фондовий ринок зберігає дані саме в такому форматі для можливості завантаження цих даних в ком'ютер.

Наша задача полягає у моделюванні прогнозування процесу, що задовольняє СДР такого виду:

$$dX_t = (\theta_1 X_t + \theta_2 X_t^2 + \theta_3 t + \theta_4 t^2)dt + (\theta_5 X_t + \theta_6 X_t^2 + \theta_7 t + \theta_8 t^2)dB_t,$$

де B_t є броунівський рух [17].

Навчання відбувається за допомогою пакету Sim.DiffProc [16] в мові програмування R. Цей пакет надає широкий спектр інструментів для моделювання, оцінки, аналізу та візуалізації стохастичних диференціальних систем в формах Іто і Стратоновича. Додатково використовуватимуться методи Ейлера-Маруями, Ейлера, Кесслера, Озакі, Шоджі-Озакі.

Знайдемо кореляцію між компаній. Для цього будемо використовувати мову програмування Python. Завантажимо дані за період з "2008-01-01" по "2009-12-31".

Будемо називати компанії сильнокорельованими, якщо значення кореляції між ними > 0.66

Stock1	Stock2	Value of correlation
L	SPG	0.660001
EBAY	PH	0.660009
EFX	MSFT	0.660014
EFX	UDR	0.660020
SIVB	TROW	0.660021
TJX	VFC	0.660022

Виберемо такі компанії після отримання результату виконання програми:

EFX (Equifax Inc.) спортовари та UDR (UDR, Inc.) – ріелторська діяльність.

Виведемо загальні дані компанії UDR за такий період часу з "2008-01-01" до "2009-12-31".

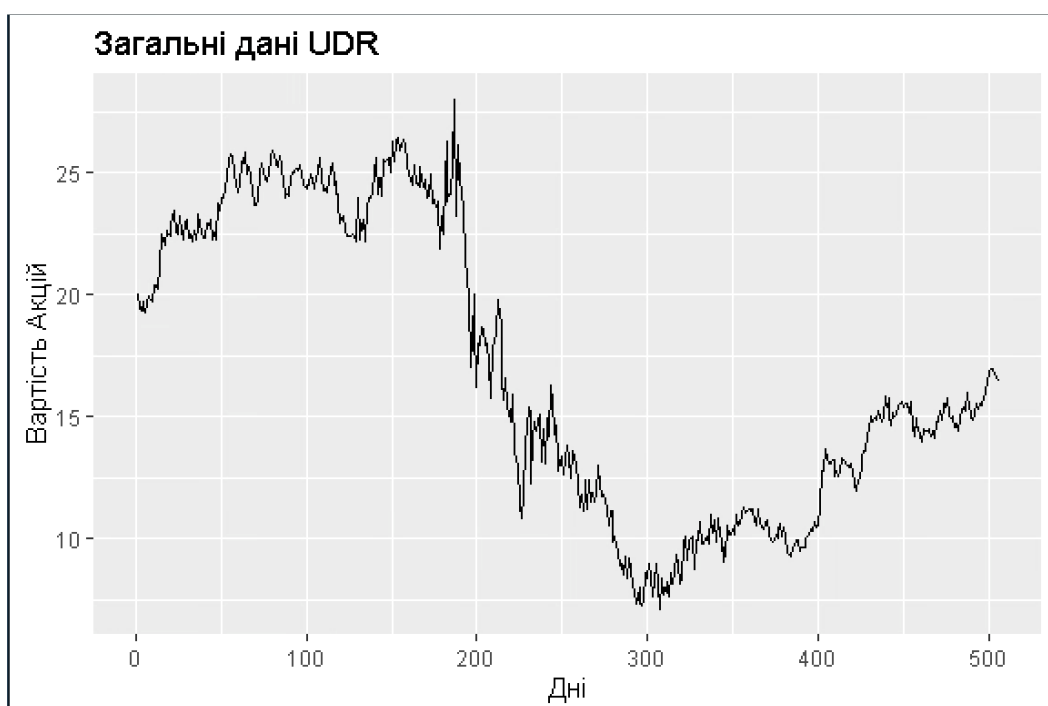


Рис. 6 загальні дані компанії UDR

Зробимо прогнозування компанії UDR за 90 днів.

Для моделювання прогнозування поділимо дані на 2 частини. Перша – це 415 днів, які будуть використовуватися для тренування нашого прогнозування. Друга – це з 416 по 506 будуть використовуватися для тесту, тобто це реальні дані, які ми будемо порівнювати з прогнозуванням, та дані, що ми отримаємо після прогнозування.

Додатково будемо використовувати методи Ейлера та Кесслера. Виберемо такі стартові параметри для СДР $\theta_1 = \theta_2 = \dots = \theta_n = 0,01$, де $n \in [1;8]$, $n \in \mathbb{N}$ на основі аналізу наявної вибірки.



Рис. 7 Прогнозування на 90 днів з використанням методу Кесслера для компанії UDR

Використаємо методи Ейлера та Шоджі-Озакі за цей же період:



Рис. 8 Прогнозування на 90 днів з використанням методу Шоджі-Озакі для компанії UDR

Використаємо методи Ейлера та Озакі за цей же період:

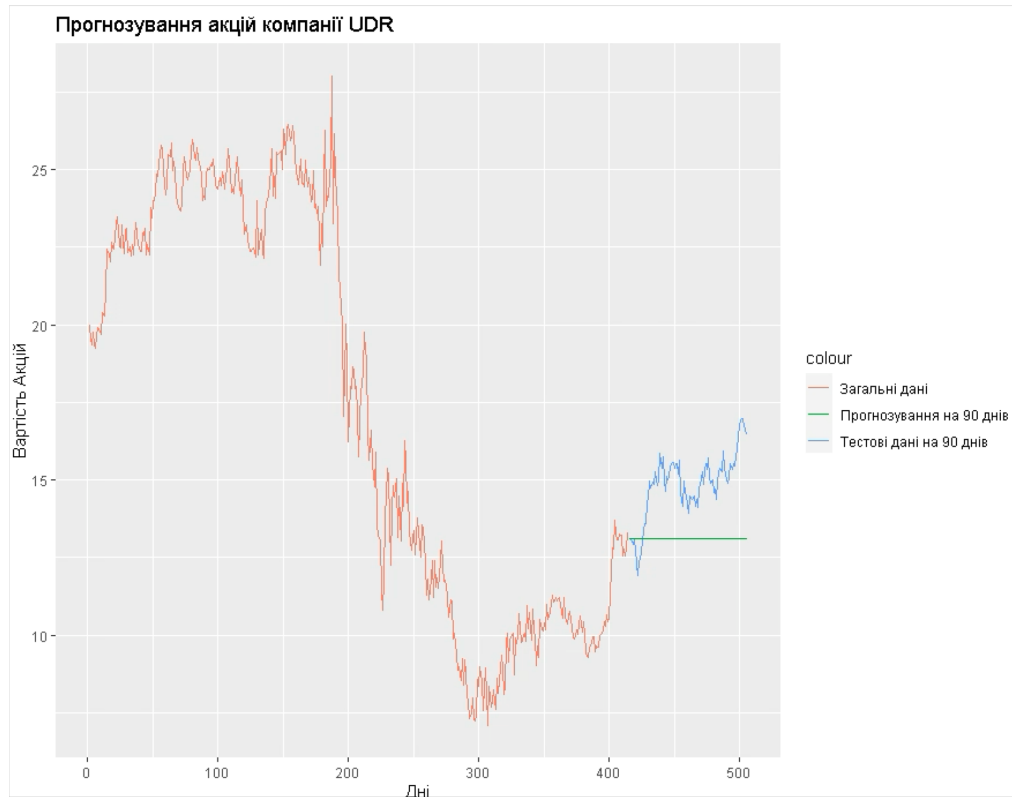


Рис. 9 Прогнозування на 90 днів з використанням методу Озакі для компанії UDR

Для покращення нашого результату почнемо відлік даних для тренування з зростаючого тренду [300,415] дні. Це дозволить отримати кращі результати в передбаченні.

Матимемо:

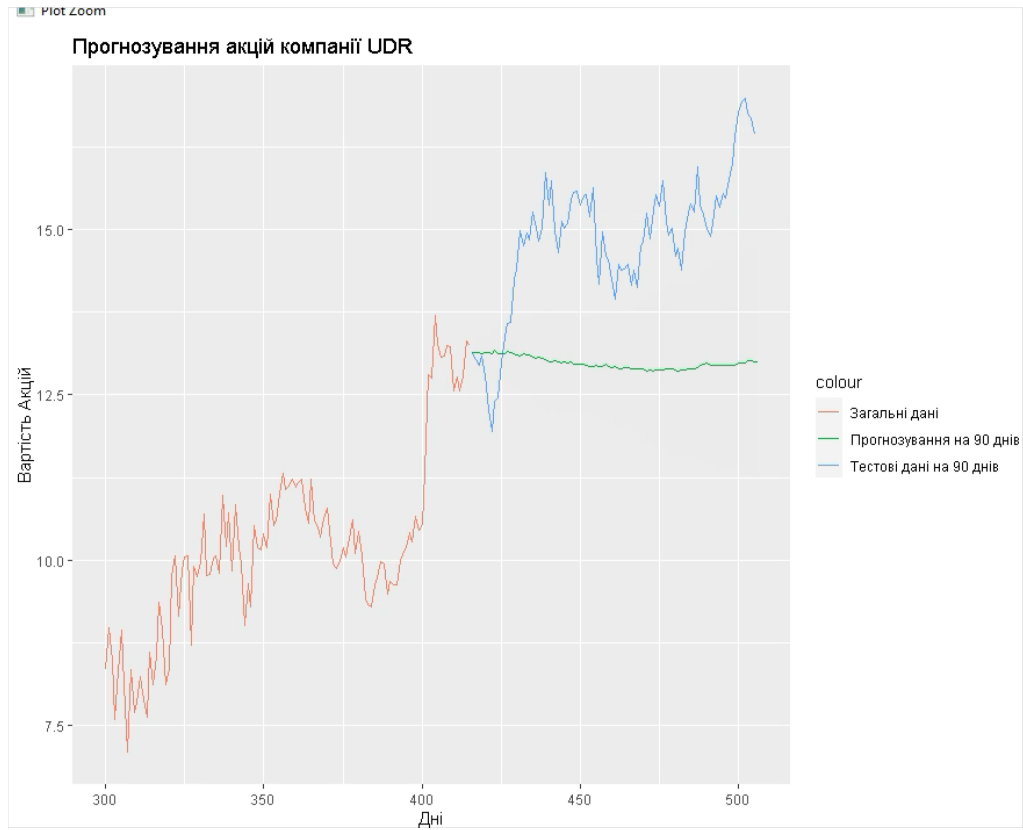


Рис. 10 Прогнозування на 90 днів з використанням методу Кесслера для компанії UDR

Змінімо параметри для СДР $\theta_1 = 5, \theta_2 = 4,5, \theta_3 = 3, \theta_4 = 3, \theta_5 = 4, \theta_6 = 2, \theta_7 = 1, \theta_8 = 3$. Вибір параметру відбувався на основі аналізу вибірки [18] [19].

Тоді:

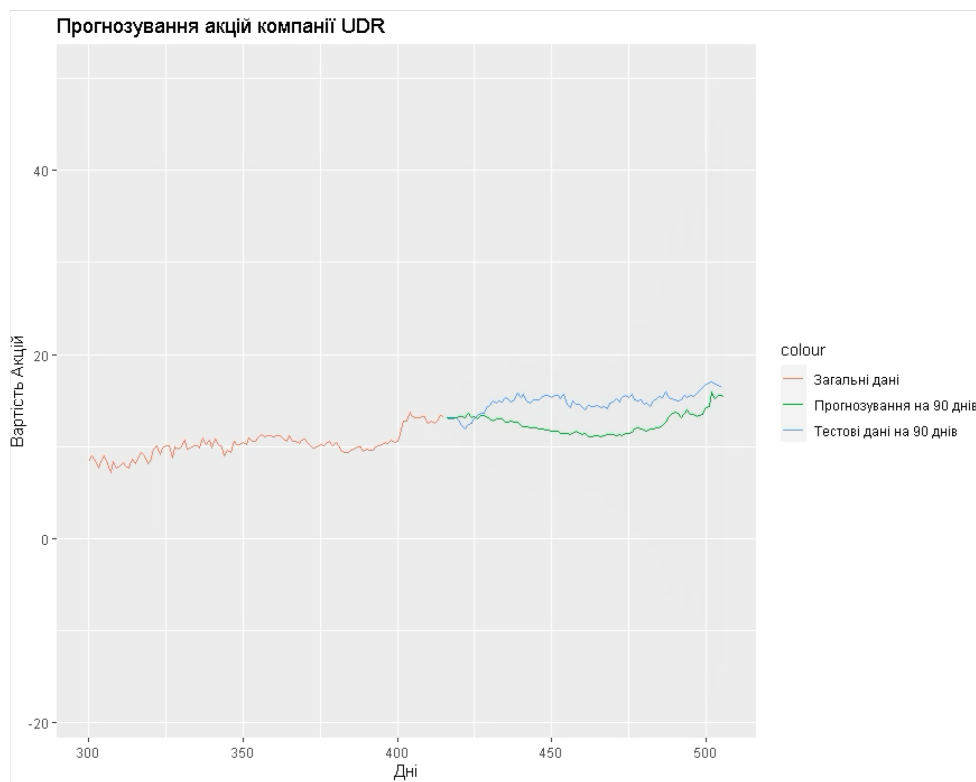


Рис. 11 Прогнозування на 90 днів з використанням методу Кесслера для компанії UDR

Розглянемо 14 денний проміжок:

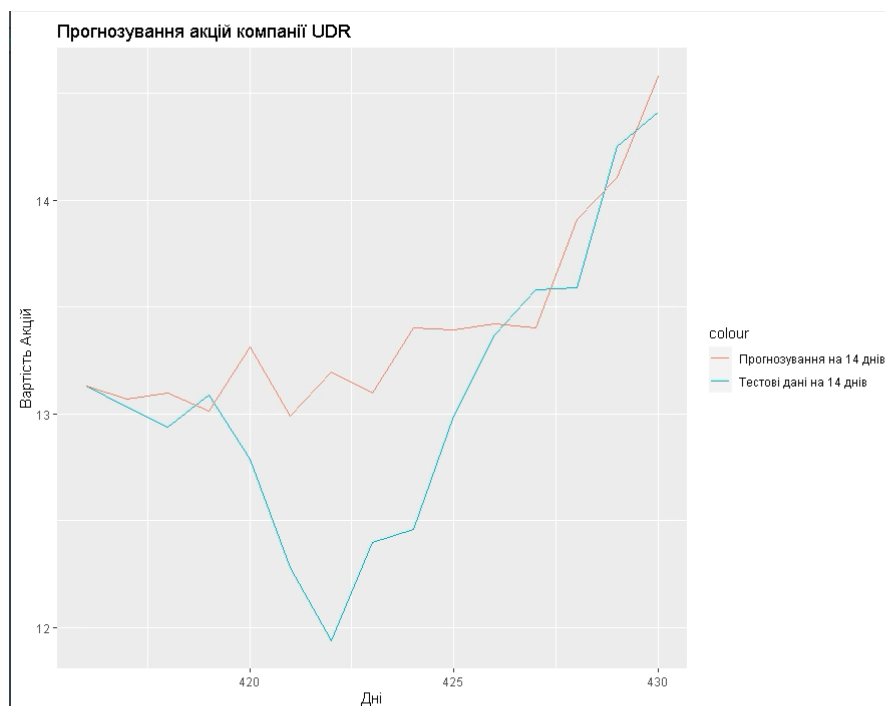


Рис. 12 Прогнозування на 14 днів з використанням методу Кесслера для компанії UDR

Як ми можемо бачити, існують закономірності, де тестові дані зростають, дані прогнозування також показують зріст. Також, коли тестові дані спадають, дані прогнозування також спадають. Для більшої точності потрібно підібрати більш кращі параметри в наш СДР.

Виведемо загальні дані компанії EFX з "2008-01-01" по "2009-12-31".



Рис. 13 EFX загальні дані

Зробимо прогнозування компаній EFX за 90 днів.

Будемо використовувати методи Кесслера та Ейлера. Початок відліку виставимо з зростаючого тренду, а саме з 300 дня, використовуватимемо такі ж параметри $\theta_1 = 5, \theta_2 = 4,5, \theta_3 = 3, \theta_4 = 3, \theta_5 = 4, \theta_6 = 2, \theta_7 = 1, \theta_8 = 3$.

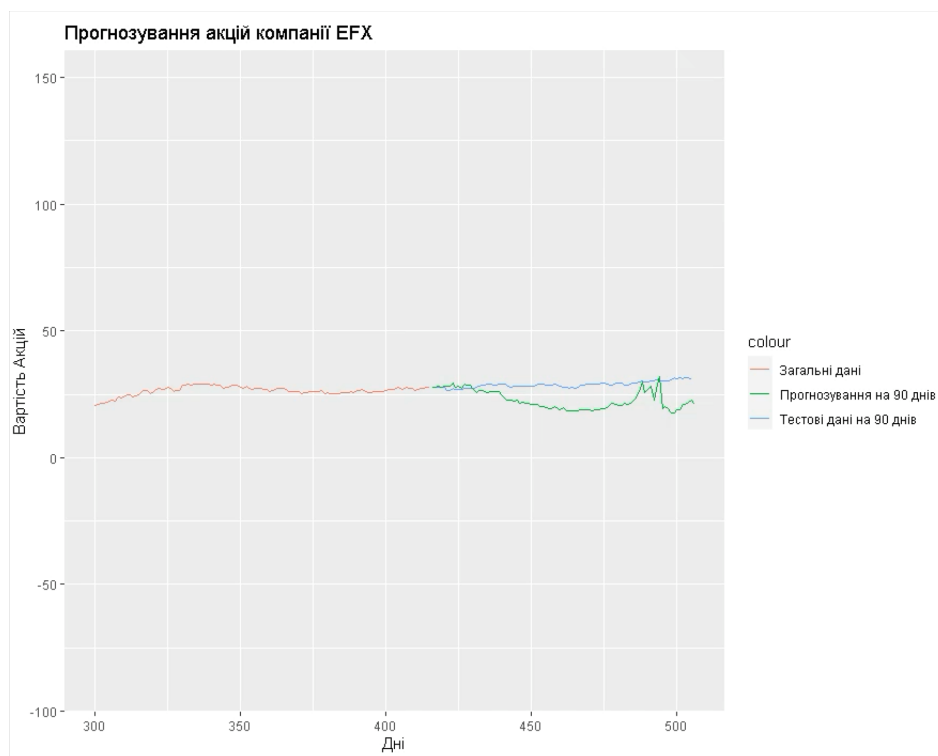


Рис. 14 Прогнозування на 90 днів з використанням методу Кесслера для компанії EFX

Розглянемо 14 денний проміжок:

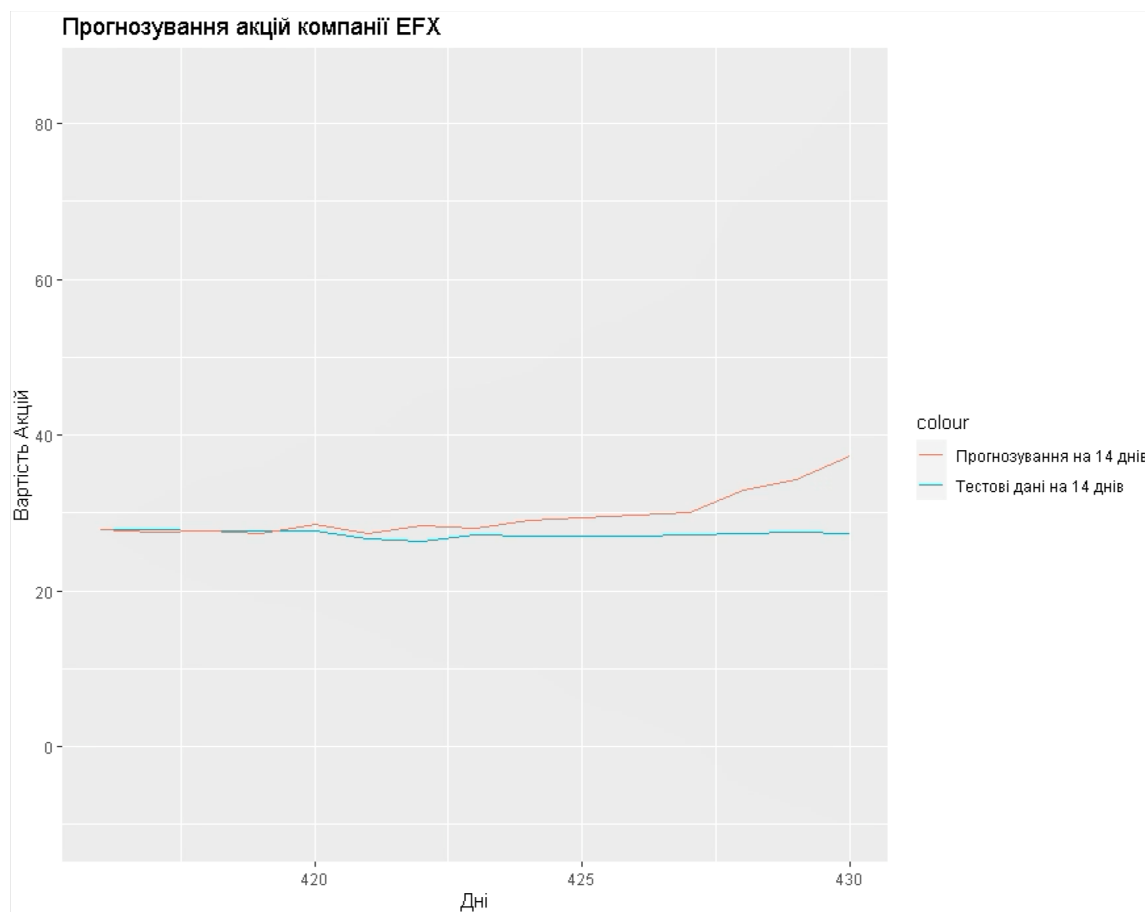


Рис. 14 Прогнозування на 14 днів з використанням методу Кесслера для компанії EFX

Ми побачили можливість побудови прогнозування різними методами та на різні проміжки часу, було виведено параметри, за якими відбувалася побудова, побачили схожість трендів на проміжках. На чим більший проміжок часу ми хочемо передбачити, тим менш точніші в нас результати.

3.2 Приклади застосування виявлених стохастичних закономірностей та методів прогнозування

Нехай у опціоні задіяні цінні папери (товари) підприємств $F_1, F_2, \dots, F_n$. Позначимо їх ціни за умови інвестицій у ціну опціону 1 грошової одиниці через $\xi_1, \xi_2, \dots, \xi_n$ індикатори збереження траєкторії цін протягом строку дії опціону у межах близької до змодельованої траєкторії, через $I_1, I_2, \dots, I_n$, індикатори знаходження цін реалізації опціону у межах окупності через $j_1, j_2, \dots, j_n$, ціни опціону, за умови виходу з близькості до змодельованої траєкторії через $\eta_1, \eta_2, \dots, \eta_n$, індикатори їх знаходження у межах окупності $j_1^*, j_2^*, \dots, j_n^*$.

Таким чином, для прибутку, отриманого в результаті реалізації опціону маємо:

$$I = \sum_{k=1}^n \alpha_k (I_k j_k \xi_k + \overline{I_k j_k^*} \eta_k) - \sum_{k=1}^n \alpha_k = \sum_{k=1}^n \alpha_k (I_k j_k \xi_k + \overline{I_k j_k^*} \eta_k - 1)$$
, де $\alpha_1, \dots, \alpha_n$ є кількість коштів, вкладених у папери (товари), відповідно, підприємств $F_1, F_2, \dots, F_n$ при купівлі опціонів.

Переходячи до математичного сподівання маємо:

$$EI = \sum_{k=1}^n \alpha_k E(I_k j_k \xi_k + \overline{I_k j_k^*} \eta_k - 1)$$

Враховучи, що

$$\alpha_i \geq 0 \quad i = \overline{1, n}$$

Найбільший очікуваний прибуток має місце у випадку, коли

$$\alpha_1 + \dots + \alpha_n = \alpha_m$$

Причому

$$\max_{1 \leq k \leq n} E(I_k j_k \xi_k + \overline{I_k j_k^*} \eta_k) = E(I_m j_m \xi_m + \overline{I_m j_m^*} \eta_m)$$

3.3 Моделювання портфеля ф'ючерсів

Для моделювання портфеля ф'ючерсів будемо використовувати мову програмування Python з використанням бібліотек pandas, numpy, random, pandas, _datareader, data, datetime, matplotlib , yfinance.

Я буду показувати наш портфель ф'ючерса з різними видатками, підняттям зарплати, інфляцією та виходом на пенсію, використовуючи задані проміжкові дані довільної компанії.

Для показу вищезазначених умов мені потрібно ввести початкові умови, які я буду відображати на графіках поступово.

Нехай наш інвестиційний щорічний вклад складає 500000 протягом 10 років.

Відображенням цієї умови є лінійна пряма:

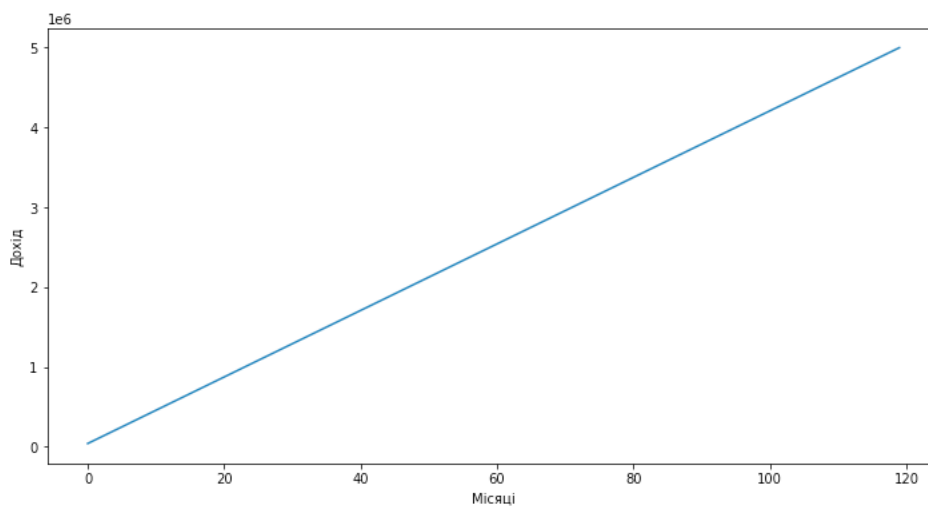


Рис. 15 Прибуток через 10 років

Введемо поняття податку на дохід, якій становитиме 20% і буде зніматися щомісячно з щорічним підняттям заробітної плати на 10%.

Матимемо таке відображення:

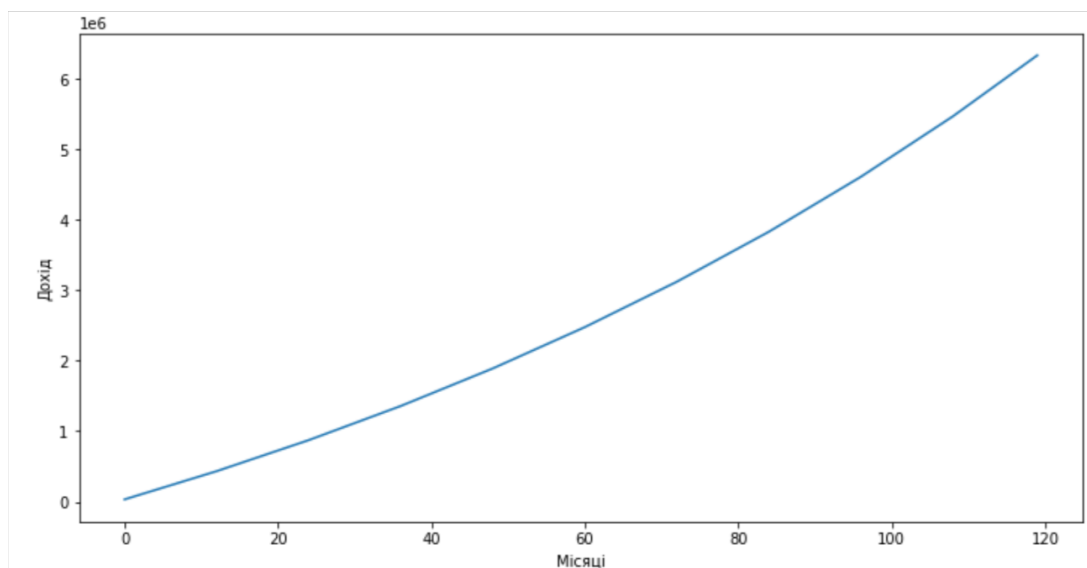


Рис. 16 Прибуток за 10 років з врахуванням оподаткування та підвищення

Почнемо виконувати моделювання прибутків. Використовуватимемо історичні дані за період з «2000-12-31» до «2021-01-02» компанії EFX.

Введемо додаткові змінні, як початкові інвестиції з сумою в 250000 та видатки:

Іпотека = 6500, оплата кредитних карток = 1750, послуги медичного страхування = 3250, внески в пенсійний фонд = 1500 та інші витрати на потреби = 15000. Зазначимо, що ці видатки повинні бути сплачені щомісяця. Для більш точнішого відображення ще потрібно ввести зміну інфляції = 8%, яка буде відображатися на наших видатках щорічно.

Для моделювання нам потрібно обчислити середньомісячну прибутковість та стандартне відхилення для вибраного нами історичного періоду, зазначимо ще те, що ми можемо легко змінити всі наші дані так як вони задані в змінних.

Обчисливши середньомісячну прибутковість та стандартне відхилення, ми їх зберігаємо, щоб потім використати при заданні вхідних даних до нормального розподілу.

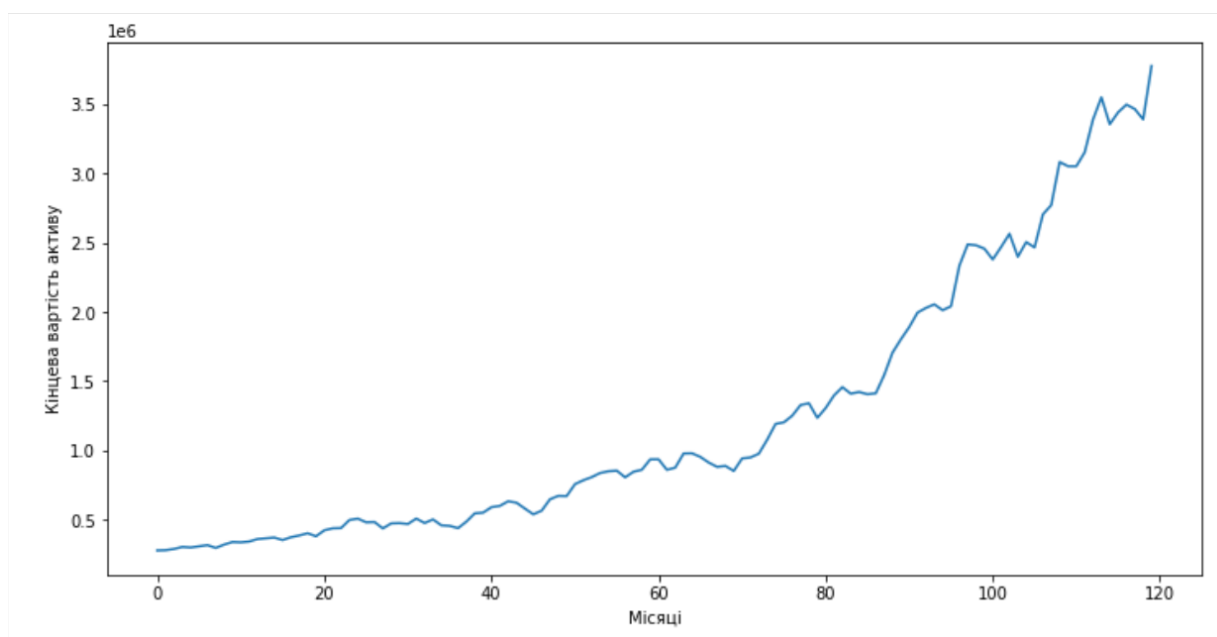


Рис. 17 Прибуток за 10 років при інвестуванні в компанію EFX

На цьому графіку ми бачимо, що інвестиційний прибуток має випадкову природу, що пояснюється тим, що один день активи цієї компанії зростали, інший – спадали.

Обчислюючи значення середньомісячного прибутку та стандартного відхилення, та ввівши значення видатків та інфляції, ми імітуємо реальні, повсякденні дані.

Встановимо дату в майбутньому, коли плануємо вийти на пенсію, тоді наш активний дохід у вигляді зарплати припиняється, якщо не зникне повністю. Змінімо виплати іпотеки на 0, так як припускатимемо, що до цього віку в нас вона буде виплаченою. Введемо значення 50 років – через скільки років ми вийдемо на пенсію.

Отримаємо:

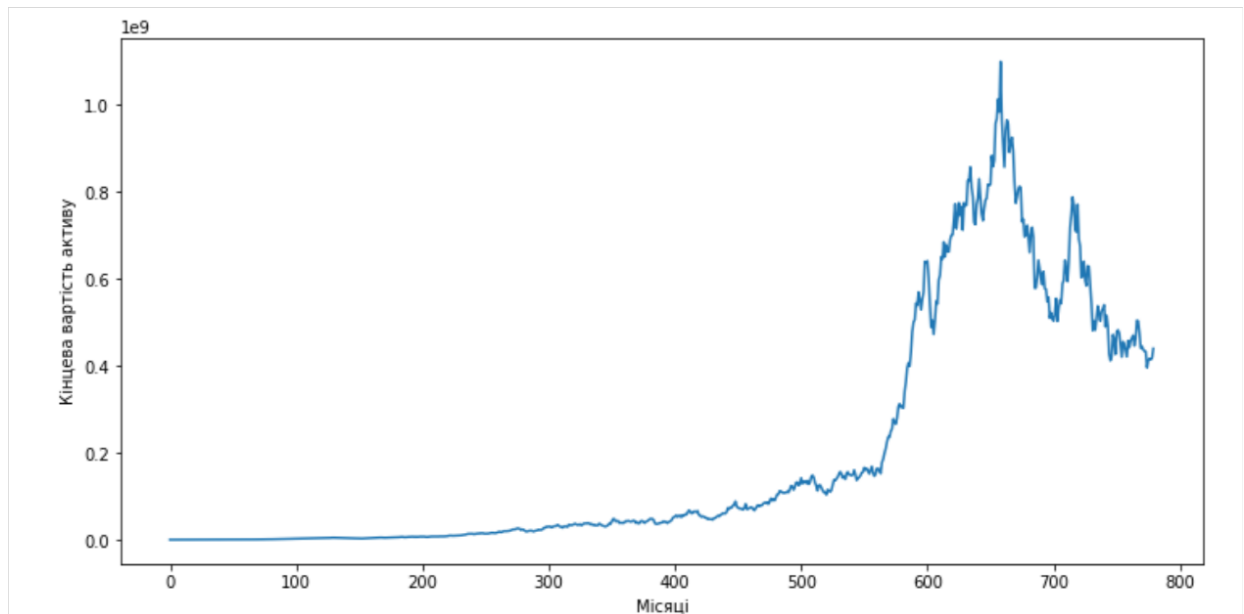


Рис. 18 Прогнозування з виходом на пенсію при інвестуванні в компанію EFX

ВИСНОВКИ

У цій роботі ми намагаємося зосередитися на можливості побудови прогнозів у фінансовій математиці за допомогою застосування стохастичних диференціальних рівнянь. Також, для роботи ми використали сучасні методи визначення структури стохастичних диференціальних рівнянь та серйозні обчислювальні потужності.

Нами проведено роботу з прогнозування показників фінансових ринків, формування портфелю ф'ючерсів з різними умовами та застосування виявлених стохастичних закономірностей та методів прогнозування.

Набутий досвід обробки літератури, впровадження комп'ютерних алгоритмів та візуалізації даних, дозволить продовжити роботу в обраному напрямку. Використання сучасних інформаційних технологій уможливило проведення цієї роботи та дозволило провести подальші дослідження.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] Ширяев А.Н. Основы стохастической финансовой математики: Т.1: Факты, модели. Электронное издание М. МЦНМО, 2016 - 440 с.
- [2] Oksendal B., Stochastic differential equations, Springer, 2000 - 332 p.
- [3] Степанов С.С. Стохастический мир, Электронное издание, 2012 - 376 с.
- [4] Гихман И.И., Скороход А.В. Стохастические дифференциальные уравнения, К.: Наукова думка, 1982 - 612с.
- [5] Guidoum AC, Boukhetala K “Performing Parallel Monte Carlo and Moment Equations Methods for Itô and Stratonovich Stochastic Differential Systems: R Package Sim.DiffProc”. Journal of Statistical Software, 96(2), 2020. 1–82.
- [6] Rodgers; Nicewander "Thirteen ways to look at the correlation coefficient, 1988: 59–66.
- [7] Oxford Dictionary of Statistics, Oxford University Press, 2002, p. 104
- [8] Kloeden, P.E. & Platen, E. *Numerical Solution of Stochastic Differential Equations. Springer, Berlin. 1992*
- [9] Aït-Sahalia, Y. "Maximum Likelihood Estimation of Discretely Sampled Diffusion: A Closed-Form Approximation Approach". *Econometrica*. 70 (1), 2002: 223–262
- [10] K. C. Chan, G. Andrew Karolyi, Francis A. Longstaff and Anthony B. Sanders “An Empirical Comparison of Alternative Models of the Short-Term Interest Rate”, 1992.
- [11] Kessler, M. (1997). Estimation of an ergodic diffusion from discrete observations. *Scand. J. Statist.*, 24, 211-229.
- [12] Iacus, S.M. (2008). *Simulation and inference for stochastic differential equations: with R examples.*
Springer-Verlag, New York.

- [13] Schwarz, Gideon E. (1978), "Estimating the dimension of a model", *Annals of Statistics*, 6 (2): 461–464
- [14] Orlando, Giuseppe; Mininni, Rosa Maria; Bufalo, Michele (2018). "A New Approach to CIR Short-Term Rates Modelling". *New Methods in Fixed Income Modeling. Contributions to Management Science*. Springer International Publishing: 35–43.
- [15] Vasicek, O. (1977). "An equilibrium characterization of the term structure". *Journal of Financial Economics*. 5 (2): 177–188.
- [16] A.G.Chouaib , K. Boukhetala.(2015), " Estimation of Stochastic Differential Equations with Sim.DiffProc Package Version 3.1 ".
- [17] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [18] Dohnal, G. (1987) "On estimating the diffusion coefficient: 105.
- [19] Bugra.A (2015) "Trend Estimation Methods for Time Series Signals.

ДОДАТОК

Виконання коду в мові програмування Python та R, розділений по розділам

Розділ 2.1 Метод Ейлера-Маруями

Додаткові бібліотеки, що дозволять провести наступні розрахунки

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

Введемо клас Моделі

Об'єктно орієнтовне програмування дозволяє спростити написання і розуміння коду

```
class Class_Moleley:
```

```
    """
```

```
    Константи стохастичної моделі.
```

```
    """
```

```
    theta_neobhidna_dlya_rorhakunku = 0.5
```

```
    mu_neobhidna_dlya_rozrahunku = 1
```

```
    sigma_dlya_rozrahunku = 0.03
```

Введемо функцію для подальшого проведення розрахунків з М'ю

```
def mu_neobhidna_dlya_rozrahunku(y: float, _t: float) -> float:
```

```
    """
```

```
    Реалізуємо м'ю Орнштейна-Уленбека для розрахунку.
```

```
    """
```

```
    return Class_Moleley.theta_neobhidna_dlya_rorhakunku *
```

```
(Class_Moleley.mu_neobhidna_dlya_rozrahunku - y)
```

Введемо функцію для подальшого проведення розрахунків з Сигмою

```
def sigma_dlya_rozrahunku(_y: float, _t: float) -> float:
```

```

"""
Впровадити сигму Орнштейна–Уленбека для розрахунку.
"""

return Class_Moleley.sigma_dlya_rozrahunku

# Введемо функцію для підбору випадкового числа
def dW(delta_t: float) -> float:
    """
    При кожній необхідності підбираємо випадкове число.
    """
    return np.random.normal(loc=0.0, scale=np.sqrt(delta_t))

# Введемо функцію для моделювання
def run_simu_neobhidna_dlya_rozrahunkulation():
    """
    Повертає результат повного моделювання.
    """
    start = 0
    end = 7
    N = 1000
    dt = float(end - start) / N
    t_start = np.arange(start, end + dt, dt)

    Y_start = 0

    ys = np.zeros(N + 1)
    ys[0] = Y_start
    for i in range(1, t_start.size):

```

```

    t = start + (i - 1) * dt
    y = ys[i - 1]
    ys[i] = y + mu_neobhidna_dlya_rozrahunku(y, t) * dt +
sigma_dlya_rozrahunku(y, t) * dW(dt)

```

```

    return t_start, ys

```

Введемо функцію для відображенні симуляції на графіках

```

def plot_simu_neobhidna_dlya_rozrahunkulations(num_sims: int) -> None:

```

```

    """

```

```

    Покажемо декілька симуляцій

```

```

    """

```

```

    for _ in range(num_sims):

```

```

        plt.plot(*run_simu_neobhidna_dlya_rozrahunkulation())

```

```

    plt.xlabel("Час")

```

```

    plt.ylabel("y")

```

```

    plt.show()

```

```

if __name__ == "__main__":

```

```

    NUM_SIMS = 50

```

```

    plot_simu_neobhidna_dlya_rozrahunkulations(NUM_SIMS)

```

Розділ 2.2 Метод Ейлера

```

set.seed(12345, kind = "L'Ecuyer-CMRG")
f_funktsiya <- expression( (1+2*x) ) ; g_funktsiya <- expression( 0.5*x^0.3 )
sumulyatsiya <-
snssdel(drift=f_funktsiya,diffusion=g_funktsiya,x0=2,N=10^4,Dt=10^-4)
dani <- sumulyatsiya$X
f_x <- expression( theta[1]+theta[2]*x ) ## дрейф моделі
g_x <- expression( theta[3]*x^theta[4] ) ## дифузія моделі
fit <- fitsde(data = dani, drift = f_x, diffusion = g_x, start = list(theta1=1,
theta2=1,
theta3=1,theta4=1),pmle="euler")
fit

```

Розділ 2.3 Метод Кесслера

```

set.seed(1234, kind = "L'Ecuyer-CMRG")
f_func <- expression(3*t*(sqrt(t)-x)) ; g_func <- expression(0.3*t)
simulation <-
snssdel(drift=f_func,diffusion=g_func,M=1,N=1000,x0=2,Dt=0.001)
datadata <- simulation$X

f_x <- expression( theta[1]*t* ( theta[2]*sqrt(t) - x ) ) ## дрейф моделі
g_x <- expression( theta[3]*t ) ## дифузія моделі
fit <- fitsde(data=datadata,drift=f_x,diffusion=g_x,start = list(theta1=1,
theta2=1,theta3=1),pmle="kessler")
summary(fit)

```

Розділ 2.4 Метод Озакі

```

set.seed(1234, kind = "L'Ecuyer-CMRG")
f_funktsiya <- expression( 3*(2-x) ) ; g_funktsiya <- expression( 0.5 )
sumulyatsiya <- snssde1d(drift=f_funktsiya,diffusion=g_funktsiya,x0=5,Dt=0.01)
danii <- sumulyatsiya$X

f_x <- expression( theta[1]*(theta[2]- x) ) ## дрейф моделі
g_x <- expression( theta[3] ) ## дифузія моделі
fit <- fitsde(data=danii,drift=f_x,diffusion=g_x,start = list(theta1=1,theta2=1,theta3=1),pmle="ozaki")
summary(fit)

```

Розділ 2.5 Метод Шоджі-Озакі

```

set.seed(1234, kind = "L'Ecuyer-CMRG")
f_func <- expression(-2*x*t) ; g_func <- expression(0.2*x)
simulation <- snssde1d(drift=f_func,diffusion=g_func,N=1000,Dt=0.001,x0=10)
data <- simulation$X

f_x <- expression( theta[1]*x*t ) ## дрейф моделі
g_x <- expression( theta[2]*x ) ## дифузія моделі
fit <- fitsde(data=data,drift=f_x,diffusion=g_x,start = list(theta1=1,
theta2=1),pmle="shoji",lower=c(-3,0),upper=c(-1,1))
summary(fit)

```

Розділ 3.1 Отримання висококорельованих компаній

#бібліотеки, які використовуватимемо для визначення кореляції

```
import numpy as np
```

```
import pandas as pd
```

```
import yfinance
```

Джерело для отримання всіх компаній з S&P 500 за потрібний нам час

```
url = "https://raw.githubusercontent.com/datasets/s-and-p-500-companies/master/data/constituents.csv"
```

```
chitannya = pd.read_csv(url)
```

```
ticker = yfinance.Ticker("SPY")
```

```
days = ticker.history(interval="1d",start="2008-01-01",end="2009-12-31").shape[0]
```

Вивід даних і перевірка за цей період.

```
total = None
```

```
for symbol in chitannya['Symbol']:
```

```
    ticker = yfinance.Ticker(symbol)
```

```
    close = ticker.history(interval="1d",start="2008-01-02",end="2010-01-01")[['Close']].pct_change(1)
```

```
    close.rename(columns={'Close':symbol},inplace=True)
```

```
    if close.shape[0] < days:
```

```
        continue
```

```
    if total is None:
```

```
        total = close
```

```
    else:
```

```
        total = total.merge(close,left_index = True, right_index = True,suffixes = ("",""))
```

```
    print(symbol,total.shape)
```

Це повний фрейм даних. Кожен стовпець – це щоденний прибуток акції, ім'я якої є назвою стовпця. Індексом є дата.

```
total
```

Обчислимо лінійну кореляцію між кожним запасом за допомогою методу «corr» Pandas,

потім ми можемо нормалізувати цю таблицю за допомогою методу «melt», щоб транспонувати стовпці в рядки.

```
corr_table = total.corr()
```

```
corr_table['stock1'] = corr_table.index
```

```
corr_table = corr_table.melt(id_vars = 'stock1', var_name =  
"stock2").reset_index(drop = True)
```

Як ви можете бачити, ми маємо кореляцію кожної акції з самим собою (яка, очевидно, дорівнює 1)

#, і ми отримуємо кореляцію запасу А з запасом В і В з А, яка однакова, оскільки кореляція є симетричною.

Ми можемо видалити ці непотрібні записи та обчислити абсолютне значення індексу кореляції.

```
corr_table
```

Відсортуємо нашу таблицю за зростанням абсолютного значення,

щоб мати менш корельовані пари акцій у перших записах.

```
corr_table = corr_table[corr_table['stock1'] < corr_table['stock2']].dropna()
```

```
#corr_table['abs_value'] = np.abs(corr_table['value'])
```

```
#highest_corr = corr_table.sort_values("abs_value",ascending = True)
```

```
first_20 = corr_table[corr_table['value']<0.33]
```

```
train_corr = first_20.sort_values("value", ascending = True).head(30)
```

І, нарешті, це таблиця менш корельованих пар акцій S&P 500 за період 2008–2010 років.

```
train_corr
```

```
# Відсортуємо нашу таблицю за зростанням абсолютного значення,
# щоб мати менш корельовані пари акцій у перших записах.
corr_table = corr_table[corr_table['stock1'] < corr_table['stock2']].dropna()
#corr_table['abs_value'] = np.abs(corr_table['value'])
#highest_corr = corr_table.sort_values("abs_value",ascending = True)
last_20 = corr_table[(corr_table['value']<=1) & (corr_table['value']>0.66)]
train_corr_last = last_20.sort_values("value", ascending = True).head(50)
train_corr_last
```

Розділ 3.1 Прогнозування компанії UDR на 90 днів

```
library(Sim.DiffProc)
library(ggplot2)aa
set.seed(314)
# Зчитуємо дискретні завантажені дані
dani <- read.table("C:/Users/admin/Documents/forecasting-
main/dani/UDR.csv", sep=',', header=T)
calculate <- dani$Close

# створюємо рівняння з параметрами
d <- expression(theta[1]*x + theta[2]*t + theta[3]*x^2 + theta[4]*t^2)
s <- expression(theta[5]*x + theta[6]*t + theta[7]*x^2 + theta[8]*t^2)

# 90 днів на створення прогнозування
calculate_test <- calculate[416:506]
# 415 днів для навчання
#calculate <- calculate[1:415]
# 115 днів для навчання
calculate <- calculate[300:415]
```

```
x_0 = calculate_test[1]
```

```
# Використання fitsde бібліотеки
```

```
mydani <- ts(calculate)
```

```
# Для отримання методу Кесслера fit <- fitsde(dani = mydani, drift = d,
diffusion = s, start = list(theta1 = 0.01, theta2 = 0.01, theta3 = 0.01, theta4 = 0.01,
theta5 = 0.01, theta6 = 0.01, theta7 = 0.01, theta8 = 0.01), pmle = "kessler")
```

```
# Для отримання методу Озакі fit <- fitsde(dani = mydani, drift = d, diffusion =
s, start = list(theta1 = 0.01, theta2 = 0.01, theta3 = 0.01, theta4 = 0.01, theta5 = 0.01,
theta6 = 0.01, theta7 = 0.01, theta8 = 0.01), pmle = "ozaki")
```

```
# Для отримання методу Шоджі-Озакі fit <- fitsde(dani = mydani, drift = d,
diffusion = s, start = list(theta1 = 0.01, theta2 = 0.01, theta3 = 0.01, theta4 = 0.01,
theta5 = 0.01, theta6 = 0.01, theta7 = 0.01, theta8 = 0.01), pmle = "shoji")
```

```
fit <- fitsde(dani = mydani, drift = d, diffusion = s, start = list(theta1 = 5, theta2 =
4.5, theta3 = 3, theta4 = 3, theta5 = 4, theta6 = 2, theta7 = 1, theta8 = 3), pmle =
"kessler")
```

```
# Створимо наші 8 параметрів
```

```
perenos1 <- as.numeric(fit$coef[1])
```

```
perenos2 <- as.numeric(fit$coef[2])
```

```
perenos3 <- as.numeric(fit$coef[3])
```

```
perenos4 <- as.numeric(fit$coef[4])
```

```
difuziya1 <- as.numeric(fit$coef[5])
```

```
difuziya2 <- as.numeric(fit$coef[6])
```

```
difuziya3 <- as.numeric(fit$coef[7])
```

```
difuziya4 <- as.numeric(fit$coef[8])
```

```
d <- eval(substitute(expression(perenos1*x + perenos2*t + perenos3*x^2 +
perenos4*t^2), list(perenos1 = perenos1, perenos2=perenos2, perenos3=perenos3,
perenos4=perenos4)))
```

```
s <- eval(substitute(expression(difuziya1*x + difuziya2*t + difuziya3*x^2 +
difuziya4*t^2), list(difuziya1=difuziya1, difuziya2=difuziya2, difuziya3=difuziya3,
difuziya4=difuziya4)))
```

```
# Кількість симуляцій
```

```
Nsim <- 100
```

```
# sum_x створює кумулятивну суму змодельованих значень для обчислення
середнього
```

```
sum_x <- rep(0, 90)
```

```
# all_x використовується для зберігання всіх змодельованих значень для
розрахунку стандартного відхилення для довірчих інтервалів
```

```
all_x <- data.frame()
```

```
#Створимо для нашого заданого числа симуляцій:
```

```
for(i in 1:Nsim)
```

```
{
```

```
# створимо нове випадкове початкове значення для кожної симуляції
```

```
rand <- as.integer(1000 * runif(1))
```

```
set.seed(rand)
```

```
# 115 днів для навчання змодельюємо SDE за методом Ейлера
```

```
X <- snssdel(N=90, x0=x_0, Dt=0.00001, drift=d, diffusion=s,
method="euler", M=1)
```

```
sum_x = sum_x + X$X
all_x = rbind(all_x, as.numeric(X$X))
}
```

```
sum_x <- sum_x / Nsim # середнє значення
```

```
sd_x <- sapply(all_x, sd) # дисперсія
```

```
mean(abs(calculate_test - sum_x))
```

```
cor(calculate_test, sum_x)
```

```
upper <- sum_x + 1.96 * sd_x
```

```
lower <- sum_x - 1.96 * sd_x
```

```
p <- ggplot()
```

```
#p <- p + geom_line(aes(x=1:415, y=calculate, color="Загальні дані"))
```

```
p <- p + geom_line(aes(x=300:415, y=calculate, color="Загальні дані"))
```

```
p <- p + geom_line(aes(x=416:506, y=calculate_test, color="Тестові 90 дні"))
```

```
p <- p + geom_line(aes(x=416:506, y=sum_x, color="Прогнозування на 90
дні"))
```

```
p <- p + ylab("Вартість Акцій")
```

```
p <- p + xlab("Дні")
```

```
p <- p + ggtitle("Прогнозування акцій компанії UDR")
```

```
p <- p + geom_ribbon(aes(x=c(416:506), y = sum_x, ymin=lower, ymax=upper),
linetype=2, alpha=0.01)
p
```

Розділ 3.1 Прогнозування компанії UDR на 14 дні

```
library(Sim.DiffProc)
library(ggplot2)
set.seed(314)

# Зчитуємо дискретні завантажені дані
dani <- read.table("C:/Users/admin/Documents/forecasting-
main/dani/UDR.csv", sep=',', header=T)
calculate <- dani$Close

# створюємо рівняння з параметрами
d <- expression(theta[1]*x + theta[2]*t + theta[3]*x^2 + theta[4]*t^2)
s <- expression(theta[5]*x + theta[6]*t + theta[7]*x^2 + theta[8]*t^2)

# 14 днів на створення Прогнозування у
calculate_test <- calculate[416:430]

# 115 днів для навчання
calculate <- calculate[300:415]

x_0 = calculate_test[1]

# Використання fitsde бібліотеки
```

```
mydani <- ts(calculate)
```

```
fit <- fitsde(dani = mydani, perenos = d, difuziya = s, start = list(theta1 = 5, theta2  
= 4.5, theta3 = 3, theta4 = 3, theta5 = 4, theta6 = 2, theta7 = 1, theta8 = 3), pmle =  
"kessler")
```

```
# Створимо наші 8 параметрів
```

```
perenos1 <- as.numeric(fit$coef[1])
```

```
perenos2 <- as.numeric(fit$coef[2])
```

```
perenos3 <- as.numeric(fit$coef[3])
```

```
perenos4 <- as.numeric(fit$coef[4])
```

```
difuziya1 <- as.numeric(fit$coef[5])
```

```
difuziya2 <- as.numeric(fit$coef[6])
```

```
difuziya3 <- as.numeric(fit$coef[7])
```

```
difuziya4 <- as.numeric(fit$coef[8])
```

```
d <- eval(substitute(expression(perenos1*x + perenos2*t + perenos3*x^2 +  
perenos4*t^2), list(perenos1 = perenos1, perenos2=perenos2, perenos3=perenos3,  
perenos4=perenos4)))
```

```
s <- eval(substitute(expression(difuziya1*x + difuziya2*t + difuziya3*x^2 +  
difuziya4*t^2), list(difuziya1=difuziya1, difuziya2=difuziya2, difuziya3=difuziya3,  
difuziya4=difuziya4)))
```

```
# Кількість симуляцій
```

```
Nsim <- 100
```

```
# sum_x створює кумулятивну суму змодельованих значень для обчислення  
середнього
```

```
sum_x <- rep(0, 14)
```

all_x використовується для зберігання всіх змодельованих значень для розрахунку стандартного відхилення для довірчих інтервалів

```
all_x <- data.frame()
```

#Створимо для нашого заданого числа симуляцій:

```
for(i in 1:Nsim)
```

```
{
```

створимо нове випадкове початкове значення для кожної симуляції

```
rand <- as.integer(1000 * runif(1))
```

```
set.seed(rand)
```

Змодельуємо SDE за методом Ейлера

```
X <- snssde1d(N=14, x0=x_0, Dt=0.00001, drift=d, diffusion=s,
method="euler", M=1)
```

```
sum_x = sum_x + X$X
```

```
all_x = rbind(all_x, as.numeric(X$X))
```

```
}
```

```
sum_x <- sum_x / Nsim # середнє значення
```

```
sd_x <- sapply(all_x, sd) # дисперсія
```

```
mean(abs(calculate_test - sum_x))
```

```
cor(calculate_test, sum_x)
```

```

upper <- sum_x + 1.96 * sd_x
lower <- sum_x - 1.96 * sd_x

p <- ggplot()
p <- p + geom_line(aes(x=416:430, y=calculate_test, color="Тестові 14 дні"))
p <- p + geom_line(aes(x=416:430, y=sum_x, color="Прогнозування на 14
дні"))
p <- p + ylab("Вартість Акцій")
p <- p + xlab("Дні")
p <- p + ggtitle("Прогнозування акцій компанії UDR")
p

```

Розділ 3.1 Прогнозування компанії EFX на 90 днів

```

library(Sim.DiffProc)
library(ggplot2)
set.seed(314)

# Зчитуємо дискретні завантажені дані
dani <- read.table("C:/Users/admin/Documents/forecasting-main/dani/EFX.csv",
sep=',', header=T)
calculate <- dani$Close

# створюємо рівняння з параметрами
d <- expression(theta[1]*x + theta[2]*t + theta[3]*x^2 + theta[4]*t^2)
s <- expression(theta[5]*x + theta[6]*t + theta[7]*x^2 + theta[8]*t^2)

# 90 днів на створення Прогнозування у

```

```
calculate_test <- calculate[416:506]
```

```
# 115 днів для навчання
```

```
calculate <- calculate[300:415]
```

```
x_0 = calculate_test[1]
```

```
# Використання fitsde бібліотеки
```

```
mydani <- ts(calculate)
```

```
fit <- fitsde(dani = mydani, drift = d, diffusion = s, start = list(theta1 = 5, theta2 =  
4.5, theta3 = 3, theta4 = 3, theta5 = 4, theta6 = 2, theta7 = 1, theta8 = 3), pmle =  
"kessler")
```

```
# Створимо наші 8 параметрів
```

```
perenos1 <- as.numeric(fit$coef[1])
```

```
perenos2 <- as.numeric(fit$coef[2])
```

```
perenos3 <- as.numeric(fit$coef[3])
```

```
perenos4 <- as.numeric(fit$coef[4])
```

```
difuziya1 <- as.numeric(fit$coef[5])
```

```
difuziya2 <- as.numeric(fit$coef[6])
```

```
difuziya3 <- as.numeric(fit$coef[7])
```

```
difuziya4 <- as.numeric(fit$coef[8])
```

```
d <- eval(substitute(expression(perenos1*x + perenos2*t + perenos3*x^2 +  
perenos4*t^2), list(perenos1 = perenos1, perenos2=perenos2, perenos3=perenos3,  
perenos4=perenos4)))
```

```
s <- eval(substitute(expression(difuziya1*x + difuziya2*t + difuziya3*x^2 +
difuziya4*t^2), list(difuziya1=difuziya1, difuziya2=difuziya2, difuziya3=difuziya3,
difuziya4=difuziya4))))
```

```
# Кількість симуляцій
```

```
Nsim <- 100
```

```
# sum_x створює кумулятивну суму змодельованих значень для обчислення
середнього
```

```
sum_x <- rep(0, 90)
```

```
# all_x використовується для зберігання всіх змодельованих значень для
розрахунку стандартного відхилення для довірчих інтервалів
```

```
all_x <- data.frame()
```

```
#Створимо для нашого заданого числа симуляцій:
```

```
for(i in 1:Nsim)
```

```
{
```

```
# створимо нове випадкове початкове значення для кожної симуляції
```

```
rand <- as.integer(1000 * runif(1))
```

```
set.seed(rand)
```

```
# Змодельюємо SDE за методом Ейлера
```

```
X <- snssde1d(N=90, x0=x_0, Dt=0.00001, drift=d, diffusion=s,
method="euler", M=1)
```

```
sum_x = sum_x + X$X
```

```
all_x = rbind(all_x, as.numeric(X$X))
}
```

```
sum_x <- sum_x / Nsim # середнє значення
sd_x <- sapply(all_x, sd) # дисперсія
```

```
mean(abs(calculate_test - sum_x))
cor(calculate_test, sum_x)
```

```
upper <- sum_x + 1.96 * sd_x
lower <- sum_x - 1.96 * sd_x
```

```
p <- ggplot()
p <- p + geom_line(aes(x=300:415, y=calculate, color="Загальні дані"))
p <- p + geom_line(aes(x=416:506, y=calculate_test, color="Тестові 90 дні"))
p <- p + geom_line(aes(x=416:506, y=sum_x, color="Прогнозування на 90
дні"))
p <- p + ylab("Вартість Акцій")
p <- p + xlab("Дні")
p <- p + ggtitle("Прогнозування акцій компанії EFX")
p <- p + geom_ribbon(aes(x=c(416:506), y = sum_x, ymin=lower, ymax=upper),
linetype=2, alpha=0.01)
p
```

Розділ 3.1 Прогнозування компанії EFX на 14 дні

```

library(Sim.DiffProc)
library(ggplot2)
set.seed(314)

# Зчитуємо дискретні завантажені дані
dani <- read.table("C:/Users/admin/Documents/forecasting-main/dani/EFX.csv",
sep=',', header=T)
calculate <- dani$Close

# створюємо рівняння з параметрами
d <- expression(theta[1]*x + theta[2]*t + theta[3]*x^2 + theta[4]*t^2)
s <- expression(theta[5]*x + theta[6]*t + theta[7]*x^2 + theta[8]*t^2)

# 14 днів на створення Прогнозування у
calculate_test <- calculate[416:430]

# 115 днів для навчання
calculate <- calculate[300:415]

x_0 = calculate_test[1]

# Використання fitsde бібліотеки
mydani <- ts(calculate)
fit <- fitsde(dani = mydani, drift = d, diffusion = s, start = list(theta1 = 5, theta2 =
4.5, theta3 = 3, theta4 = 3, theta5 = 4, theta6 = 2, theta7 = 1, theta8 = 3), pmle =
"kessler")

```

Створимо наші 8 параметрів

```
perenos1 <- as.numeric(fit$coef[1])
```

```
perenos2 <- as.numeric(fit$coef[2])
```

```
perenos3 <- as.numeric(fit$coef[3])
```

```
perenos4 <- as.numeric(fit$coef[4])
```

```
difuziya1 <- as.numeric(fit$coef[5])
```

```
difuziya2 <- as.numeric(fit$coef[6])
```

```
difuziya3 <- as.numeric(fit$coef[7])
```

```
difuziya4 <- as.numeric(fit$coef[8])
```

```
d <- eval(substitute(expression(perenos1*x + perenos2*t + perenos3*x^2 +
perenos4*t^2), list(perenos1 = perenos1, perenos2=perenos2, perenos3=perenos3,
perenos4=perenos4)))
```

```
s <- eval(substitute(expression(difuziya1*x + difuziya2*t + difuziya3*x^2 +
difuziya4*t^2), list(difuziya1=difuziya1, difuziya2=difuziya2, difuziya3=difuziya3,
difuziya4=difuziya4)))
```

Кількість симуляцій

```
Nsim <- 100
```

sum_x створює кумулятивну суму змодельованих значень для обчислення середнього

```
sum_x <- rep(0, 14)
```

all_x використовується для зберігання всіх змодельованих значень для розрахунку стандартного відхилення для довірчих інтервалів

```
all_x <- data.frame()
```

```

#Створимо для нашого заданого числа симуляцій:
for(i in 1:Nsim)

{
  # створимо нове випадкове початкове значення для кожної симуляції
  rand <- as.integer(1000 * runif(1))
  set.seed(rand)

  # Змоделюємо SDE за методом Ейлера
  X <- snssde1d(N=14, x0=x_0, Dt=0.00001, drift=d, diffusion=s,
method="euler", M=1)

  sum_x = sum_x + X$X
  all_x = rbind(all_x, as.numeric(X$X))
}

sum_x <- sum_x / Nsim # середнє значення
sd_x <- sapply(all_x, sd) # дисперсія

mean(abs(calculate_test - sum_x))
cor(calculate_test, sum_x)

upper <- sum_x + 1.96 * sd_x
lower <- sum_x - 1.96 * sd_x

p <- ggplot()

```

```

p <- p + geom_line(aes(x=416:430, y=calculate_test, color="Тестові 14 дні"))
p <- p + geom_line(aes(x=416:430, y=sum_x, color="Прогнозування на 14
дні"))
p <- p + ylab("Вартість Акцій")
p <- p + xlab("Дні")
p <- p + ggtitle("Прогнозування акцій компанії EFX")
p <- p + geom_ribbon(aes(x=c(416:430), y = sum_x, ymin=lower, ymax=upper),
linetype=2, alpha=0.01)
p

```

Розділ 3.3 Прибуток за 10 років

```

"""Імпортування сторонніх бібліотек"""

```

```

import pandas as pd
import numpy as np
import random
import pandas_datareader.data as web
import datetime
import matplotlib as mpl
import matplotlib.pyplot as plt
import yfinance as yf
from pandas_datareader import data as pdr
%matplotlib inline

```

```

"""Параметри графіка"""

```

```

plt.rcParams['figure.figsize'] = (12,6)
# Прибуток 500000
inflows = {'dohid_za_rik':500_000}
# Початок з 01/01/2020 прогнозування на 10 років

```

```

var = {'start_date' : "01/01/2020",
       'years': 10}
pributok = []
# введення поняття місяця
months = var['years'] * 12
for month in range(months):
    income = inflows['dohid_za_rik'] / 12
    # Запис нашого прибутку через кожен рік
    pributok.append(income)
plt.plot(pd.Series(pributok).cumsum())
plt.xlabel('Місяці')
plt.ylabel('Дохід')
plt.show()

```

Розділ 3.3 Прибуток за 10 років з врахуванням підвищення та оподаткування

```

"""Підняття зарплати"""
inflows = {'dohid_za_rik':500_000}
var = {'start_date' : "01/01/2020",
       'years': 10,
       'podatok': 0.2,
       'pidnyattya_zarplati':0.1} # додаємо до нашого річного прибутку
вказані%
pributok = []
months = var['years'] * 12

for month in range(months):
    income = (inflows['dohid_za_rik'] * (1 - var['podatok'])) / 12

```

```
pributok.append(income)
# Кожні 12 місяців - ми збільшуємо нашу зарплату
if (month % 12 == 0) and (month > 0): # Умова, щоб не підняти собі ж
зарплату першого місяця
    inflows['dohid_za_rik'] *= (1 + (var['pidnyattya_zarplati']))
```

```
plt.plot(pd.Series(pributok).cumsum())
plt.xlabel('Місяці')
plt.ylabel('Дохід')
plt.show()
```

Розділ 3.3 Прибуток за 10 років при вкладанні в компанію EFX

```
# Встановимо реплікацію результатів
np.random.seed(seed=7)
# "EFX завантаження даних"
start, end = datetime.datetime(2000, 12, 31), datetime.datetime(2022, 1, 1)
yf.pdr_override()
sp = pdr.get_data_yahoo("EFX", start=start, end=end)
# розрахування середньомісячної прибутковості і стандартного відхилення
E = sp.resample('M').last().pct_change().mean().values[0]
std = sp.resample('M').last().pct_change().std().values[0]

inflows = {'dohid_za_rik':500_000,
           'starting_money': 250_000} # Початкові гроші, які будуть інвестовані

vidatki = {'rent':6500,
           'card':1750,
           'med_in':3250,
           'pension':1500,
```

```
'misc': 15000}
```

```
var = {'start_date' : "01/01/2000",
```

```
      'years': 10,
```

```
      'podatok': 0.2,
```

```
      'pidnyattya_zarplati':0.1,
```

```
      'inflation': 0.08,
```

```
      'podatok_na_pributyok': 0.2,
```

```
      'market_E': E, # Середнє значення прибутковості
```

```
      'market_std': std} # Середнє значення волатильності
```

```
pributok = []
```

```
investment_gains_storage = [] # Зберігаємо наші інвестиції
```

```
# Список для зберігання вартості активів кожного періоду
```

```
income_start = [inflows['starting_money']] # Вводимо, що переше значення -  
початкові гроші
```

```
ending_list = [] # Список для зберігання вартості активів на кінець кожного  
періоду
```

```
months = var['years'] * 12
```

```
flag_broke = False
```

```
for month in range(months):
```

```
# Перевірку 1 місяця.
```

```
    if ending_list:
```

```
        income_start.append(ending_list[-1])
```

```
    assets = income_start[-1]
```

```
    assets -= (vidatki['rent'] + vidatki['card'] + \
```

```
              vidatki['med_in'] + vidatki['pension'] + \
```

```

vidatki['misc'])

if assets <= 0:
    inv_gain = 0
    flag_broke = True
    break

# генерую щомісячну ринкову прибутковість, використовуючи
нормальний розподіл
gener = np.random.normal(var['market_E'],
                          var['market_std'],
                          1)[0]

# Прибуток Інвестицій
pributok_investitsiy = (assets * gener) * (1 - var['podatok_na_pributyok'])
# Додавляю прибуток інвестицій в список
investment_gains_storage.append(pributok_investitsiy)

assets += pributok_investitsiy

income = (inflows['dohid_za_rik'] * (1 - var['podatok'])) / 12
pributok.append(income)

if (month % 12 == 0):
    inflows['dohid_za_rik'] *= (1 + (var['pidnyattya_zarplati']))
    vidatki['rent'] *= (1 + (var['inflation']))
    vidatki['card'] *= (1 + (var['inflation']))
    vidatki['med_in'] *= (1 + (var['inflation']))
    vidatki['pension'] *= (1 + (var['inflation']))

```

```
vidatki['misc'] *= (1 + (var['inflation']))
```

```
assets += income
```

```
assets_ending = assets
```

```
# Додаю в список вартість активів на кінець періоду
```

```
ending_list.append(assets_ending)
```

```
plt.plot(pd.Series(ending_list))
```

```
plt.xlabel('Місяці')
```

```
plt.ylabel('Кінцева вартість активу')
```

```
plt.show()
```

Розділ 3.3 Прибуток при вкладанні в компанію EFX з врахуванням виходу на пенсію

```
np.random.seed(seed=7)
```

```
start, end = datetime.datetime(2000, 12, 31), datetime.datetime(2021, 1, 1)
```

```
yf.pdr_override()
```

```
sp = pdr.get_data_yahoo("EFX", start=start, end=end)
```

```
E = sp.resample('M').last().pct_change().mean().values[0]
```

```
std = sp.resample('M').last().pct_change().std().values[0]
```

```
inflows = {'dohid_za_rik': 500_000,
```

```
           'starting_money': 250_000}
```

```
vidatki = {'rent': 6500,
```

```
           'card': 1750,
```

```
           'med_in': 3250,
```

```

        'pension':1500,
        'misc': 15000}

var = {'start_date' : "01/01/2022",
        'years': 65,
        'retirement': 50,
        'podatok': 0.2,
        'pidnyattya_zarplati':0.1,
        'inflation': 0.08,
        'podatok_na_pributyok': 0.2,
        'market_E': E,
        'market_std': std}

pributok = []
investment_gains_storage = []

income_start = [inflows['starting_money']]
ending_list = []
months = var['years'] * 12

flag_broke = False

for month in range(months):

    if ending_list:
        income_start.append(ending_list[-1])

    assets = income_start[-1]
```

```

assets -= (vidatki['rent'] + vidatki['card'] + \
          vidatki['med_in'] + vidatki['pension'] + \
          vidatki['misc'])

if assets <= 0:
    inv_gain = 0
    flag_broke = True
    break

gener = np.random.normal(var['market_E'],
                        var['market_std'],
                        1)[0]

pributok_investitsiy = (assets * gener) * (1 - var['podatok_na_pributyok'])

investment_gains_storage.append(pributok_investitsiy)

assets += pributok_investitsiy

# Якщо ми пішли на пенсію - ми не отримаємо зарплати
if month >= var['retirement'] * 12:
    income = 0
else:
    income = (inflows['dohid_za_rik'] * (1 - var['podatok'])) / 12
    pributok.append(income)

if (month % 12 == 0):
    inflows['dohid_za_rik'] *= (1 + (var['pidnyattya_zarplati']))

```

```
vidatki['rent'] *= (1 + (var['inflation']))  
vidatki['card'] *= (1 + (var['inflation']))  
vidatki['med_in'] *= (1 + (var['inflation']))  
vidatki['pension'] *= (1 + (var['inflation']))  
vidatki['misc'] *= (1 + (var['inflation']))
```

```
assets += income  
assets_ending = assets  
ending_list.append(assets_ending)
```

```
plt.plot(pd.Series(ending_list))  
plt.xlabel('Місяці')  
plt.ylabel('Кінцева вартість активу')  
plt.show()
```