

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧОРНОМОРСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ПЕТРА МОГИЛИ

**ТАНЗІ САНА**

УДК 004.89

**Інформаційна система прогнозування фінансових показників на основі моделі  
ARIMA**

124 – Системний аналіз

Автореферат  
магістерської кваліфікаційної роботи на здобуття освітньої кваліфікації  
«Магістр системного аналізу»

Миколаїв – 2021

Магістерська кваліфікаційна робота є рукопис.

Робота виконана в Чорноморському національному університеті імені Петра Могили Міністерства освіти і науки України на кафедрі інтелектуальних інформаційних систем

Науковий керівник: д. т. н., професор, професор кафедри інтелектуальних інформаційних систем  
Гожий Олександр Петрович

Рецензент: к.т.н., доцент, доцент кафедри інженерії програмного забезпечення  
Солобуто Лариса Вадимівна

Захист відбудеться «25» лютого 2021 р. о 9<sup>30</sup> год. на засіданні екзаменаційної комісії (ауд. 2-403) у Чорноморському національному університеті імені Петра Могили за адресою: 54003, м. Миколаїв, вул. 68-ми Десантників, 10.

З магістерською кваліфікаційною роботою можна ознайомитися в бібліотеці Чорноморського національного університету імені Петра Могили за адресою: 54003, м. Миколаїв, вул. 68-ми Десантників, 10.

Автореферат представлений «18» лютого 2021 р.

Секретар  
екзаменаційної комісії,  
к.пед.н., доцент

Н. М. Болубаш

## ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

*Актуальність* дослідження визначається складністю прогнозування та прийняття рішень в умовах нестабільності фінансового ринку, складністю виділення найбільш важливих та впливових критеріїв для подальшого прийняття рішень.

*Метою* магістерської кваліфікаційної роботи є дослідження та реалізація прогнозування фінансових показників на основі моделі ARIMA .

*Об'єктом* дослідження є моделі прогнозування на основі дослідження часових рядів.

*Предметом* дослідження є методи прогнозування фінансових показників на основі моделей багатокритерійного вибору і аналізу.

*Практичне значення* даної магістерської кваліфікаційної роботи полягає у можливості застосування інтегрованої моделі і методології аналізу часових рядів для здійснення фінансової діяльності.

Результати даної магістерської кваліфікаційної роботи було надруковано у тезах XXIV Всеукраїнська науково-практична конференція молодих вчених, аспірантів і студентів «Інтелектуальні інформаційні системи» у секції Інформаційні системи та їх інтелектуалізація.

Магістерська кваліфікаційна робота складається із вступу, 6 розділів, висновків, додатків. Загальний обсяг роботи складає 105 сторінок, 26 рисунків, 4 таблиць та 25 посилань на літературні джерела.

## ОСНОВНИЙ ЗМІСТ РОБОТИ

У вступі магістерської кваліфікаційної роботи обґрунтовано актуальність обраної теми, сформульовано мету і задачі дослідження, визначено предмет та об'єкт дослідження.

У першому розділі наведено огляд предметної області та теоретичних засад поняття стартап-проекту в сфері інформаційних технологій. Аналіз існуючих методів і підходів показав, що перспективним напрямом у вирішенні задачі оцінювання та подальшого вибору startup-проектів є комбінування класичних методів прийняття рішень з методами і підходами нечіткої логіки. З урахуванням проведеного аналізу сформовано постановку задачі.

У другому розділі розглянута мова R як засіб ілюстрації роботи з часовими рядами, проведено дослідження способів побудови та представлення часових рядів, також перераховані та продемонстровані приклади типових операції над часовими рядами.

Розглянемо роботу з часовими рядами на основі конкретних прикладів та задач. Почнемо з задачі представлення часового ряду. Уявімо, що користувачу потрібна структура даних R, яка може представляти дані.

У цьому випадку рекомендуються пакети zoo і xts. Вони визначають структуру даних часових рядів і містять безліч корисних функцій для роботи з даними часових рядів. Необхідно створити об'єкт zoo таким чином, де  $x$  - це вектор, матриця або таблиця даних, а  $dt$  - вектор відповідних дат представлених в форматі дати і часу:

```
library(zoo)
```

```
ts <- zoo(x, dt)
```

Та створимо об'єкт xts:

```
library(xts)
```

```
ts <- xts(x, dt)
```

Наступним кроком до виконання задачі необхідно виконати перетворення між представленнями даних часового ряду, використовуючи функції `as.zoo` та `as.xts`:

`as.zoo(ts)` – перетворить `ts` в об'єкт `zoo`

`as.xts(ts)` - перетворить `ts` в об'єкт `xts`.

В R є щонайменше вісім різних реалізацій структур даних для представлення часових рядів. Можна сказати, що `zoo` і `xts` є відмінними пакетами для роботи з даними часових рядів і вони краще, ніж ті, що ми пробували.

Ці уявлення припускають, що у користувача є два вектори: вектор спостережень (дані) і вектор дат або часу цих спостережень. Функція `zoo` об'єднує їх в об'єкт `zoo`:

```
library(zoo)
#>
#> Attaching package: 'zoo'
#> The following objects are masked from 'package:base':
#>
#> as.Date, as.Date.numeric
x <- c(3, 4, 1, 4, 8)
dt <- seq(as.Date("2018-01-01"), as.Date("2018-01-05"), by = "days")
ts <- zoo(x, dt) print(ts)
#> 2018-01-01 2018-01-02 2018-01-03 2018-01-04 2018-01-05
#> 3 4 1 4 8
```

Функція `xts` діє схожим чином, вертаючи об'єкт `xts`:

```
library(xts)
#>
#> Attaching package: 'xts'
#> The following objects are masked from 'package:dplyr':
#>
#> first, last
ts <- xts(x, dt) print(ts)
#> [, 1]
#> 2018-01-01 3
#> 2018-01-02 4
```

```
#> 2018-01-03 1
```

```
#> 2018-01-04 4
```

```
#> 2018-01-05 8
```

Дані, `x`, повинні бути числовими. Вектор дат або дат і часу, `dt`, називається індексом. Допустимі індекси відрізняються в залежності від пакета:

- `zoo` – Індексом можуть бути будь-які впорядковані значення, такі як об'єкти `Date`, об'єкти `POSIXct`, цілі числа або навіть значення з плаваючою комою,
- `xts` – Індекс повинен підтримуватися класом дати або часу. Сюди входять об'єкти `Date`, `POSIXct` і `chron`. Їх повинно бути достатньо для більшості додатків, але також можна використовувати об'єкти `yearmon`, `yearqtr` і `date- Time`. Пакет `xts` більш суворий, ніж `zoo`, оскільки він реалізує потужні операції, які вимагають індексу на базі часу.

У наступному прикладі створюється об'єкт `zoo`, який містить ціну акцій IBM за перші п'ять днів 2010 року; в ньому використовуються об'єкти `Date` для індексу:

```
prices <- c(132.45, 130.85, 130.00, 129.55, 130.85)
```

```
dates <- as.Date(c(
  "2010-01-04", "2010-01-05", "2010-01-06",
  "2010-01-07", "2010-01-08"
```

```
))
```

```
ibm.daily <- zoo(prices, dates) print(ibm.daily)
```

```
#> 2010-01-04 2010-01-05 2010-01-06 2010-01-07 2010-01-08
```

```
#> 132 131 130 130 131
```

У наступному прикладі, навпаки, показана ціна на акції IBM з інтервалом в одну секунду. Він являє час за кількістю годин після опівночі, починаючи з 9:30 ранку. (1 секунда = 0,00027778 години, більш-менш):

```
prices <- c(131.18, 131.20, 131.17, 131.15, 131.17)
```

```
seconds <- c(9.5, 9.500278, 9.500556, 9.500833, 9.501111)
```

```
ibm.sec <- zoo(prices, seconds) print(ibm.sec)
```

```
#> 10 10 10 10 10
```

```
#> 131 131 131 131 131
```

У цих двох прикладах використовувався один часовий ряд, в якому дані були взяті з вектора. І zoo, і xts можуть обробляти кілька паралельних часових рядів. Для цього запишіть кілька часових рядів в матрицю або таблицю даних, а потім створіть багатовимірний часовий ряд, викликавши функцію zoo (або xts):

```
ts <- zoo(df, dt) # OR: ts <- xts(dfrm, dt)
```

Другий аргумент - це вектор дат (або дат і часу) кожного спостереження. Існує тільки один вектор дат для всіх часових рядів; іншими словами, всі спостереження в кожному рядку матриці або таблиці даних повинні мати одну і ту ж дату. Як тільки дані захоплені всередині об'єкта zoo або xts, є можливість витягти чисті дані за допомогою функції coredata, яка повертає простий вектор (або матрицю):

```
coredata (ibm.daily)
```

```
#> [1] 132 131 130 130 131
```

Витягти дату або час можна за допомогою функції index:

```
index (ibm.daily)
```

```
#> [1] "2010-01-04" "2010-01-05" "2010-01-06" "2010-01-07" "2010-01-08"
```

Пакет xts дуже схожий на zoo. Він оптимізований по швидкості, тому особливо добре підходить для обробки великих обсягів даних, а також добре справляється з перетворенням в інші уявлення тимчасового ряду і назад. Одною з великих переваг фіксування даних всередині об'єкта zoo або xts є те, що користувач отримує доступ до спеціальних функцій виведення, побудови графіків, обчислення різниць, злиття, періодичної вибірки, а також таких функцій, як rollapply, і іншим корисним операціями. Існує навіть функція read.zoo, призначена для читання даних часових рядів з файлів ASCII.

Слід пам'ятати, що пакет xts може робити все те, що може робити пакет zoo, тому всюди, де в цьому розділі йдеться про об'єкти zoo, також можна використовувати об'єкти xts.

Ці пакети встановлені на комп'ютері, їх документацію можна переглянути за допомогою функції vignette:

```
vignette ("zoo")
```

```
vignette ("xts")
```

Пакет `timeSeries` - ще одна гідна реалізація об'єкта часових рядів. Він є частиною проекту з навчання інформатиці фінансів `Rmetrics`.

Розглянемо наступну задачу – побудова одного або декількох часових рядів. Для вирішення подібної задачі пропонується використати функцію `plot(x)`, яка працює з об'єктами `zoo` і `xts`, які в свою чергу містять один або кілька часових рядів. Для простого вектора `v` спостережень часового ряду можна використовувати `plot(v, type = "l")` або `plot.ts(v)`.

У функції `plot` є версія для об'єктів `zoo` і `xts`. Вона може будувати об'єкти, які містять один або кілька часових рядів. В останньому випадку вона може побудувати кожен ряд на окремому графіку або все ряди разом на одному.

Припустимо, що `ibm.infl` - це об'єкт `zoo`, який містить два тимчасових ряду. Один показує котирувальну ціну акцій IBM з січня 2000 року по грудень 2017 року, а другий – ту ж саму ціну з поправкою на інфляцію. Якщо завдасти об'єкт на графік, R побудує два тимчасових ряду разом на одному графіку, як показано на Рис. 1.1.:

```
load(file = "./data/ibm.rdata") library(xts)
main <- "IBM: Historical vs. Inflation-Adjusted" lty <- c("dotted", "solid")
# Plot the xts object
plot(ibm.infl,
lty = lty, main = main, legend.loc = "left"
)
```

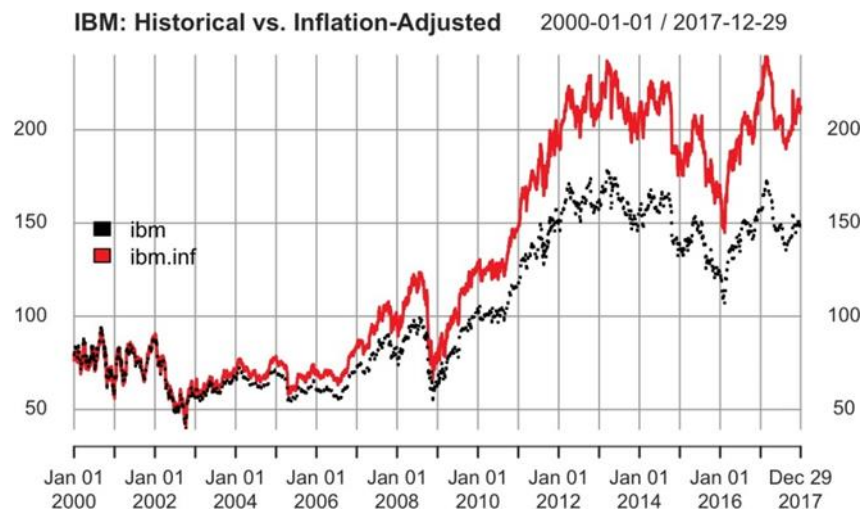


Рис.1.1 Приклад графіку `xts`



Функція `plot` для `xts` надає заголовок за замовчуванням просто як ім'я об'єкта `xts`. Як показано тут, звичайно основний параметр має значніший заголовок. Код визначає два типи ліній (`lty`), тому обидві лінії намальовані в двох різних стилях, щоб їх було легше розрізняти. Для роботи з фінансовими даними пакет `quantmod` містить спеціальні функції побудови графіків, які створюють гарні стилізовані діаграми.

У випадку якщо завдання стоїть таким чином, що користувач хоче побачити тільки найстаріші або найостанніші спостереження вашого часового ряду необхідно використовувати функцію `head`, щоб переглянути найстаріші спостереження: `head(ts)` та функцію `tail`, щоб переглянути найостанніші спостереження:

Функції `head` і `tail` є загальними, тому вони будуть працювати незалежно від того, чи зберігаються дані в простому векторі, об'єкті `zoo` або об'єкті `xts`. Припустимо, у користувача є об'єкт `xts` з багаторічною історією ціни акцій ІВМ, такий же, як в попередньому рецепті. Не можна відобразити весь набір даних, тому що він буде виходити за межі екрану. Але ви можете переглянути початкового спостереження і переглянути фінальні:

```
ibm <- ibm.infl $ ibm # Беремо одну колонку в якості ілюстрації.
```

```
head(ibm)
```

```
#> Ibm
```

```
#> 2000-01-01 78.6
```

```
#> 2000-01-03 82.0
```

```
#> 2000-01-04 79.2
```

```
#> 2000-01-05 82.0
```

```
#> 2000-01-06 80.6
```

```
#> 2000-01-07 80.2
```

```
tail(ibm)
```

```
#> Ibm
```

```
#> 2017-12-21 148
```

```
#> 2017-12-22 149
```

```
#> 2017-12-26 150
```

```
#> 2017-12-27 150
```

```
#> 2017-12-28 151
```

```
#> 2017-12-29 150
```

За замовчуванням `head` і `tail` показують (відповідно) шість найстаріших і шість останніх спостережень.

Можна побачити більше спостережень, надавши другий аргумент, наприклад `tail(ibm, 20)`. Пакет `xts` також включає в себе функції `first` і `last`, які використовують календарні періоди замість кількості спостережень. Можна використовувати ці функції для вибору даних за кількістю днів, тижнів, місяців або навіть років:

```
first(ibm, "2 week")
```

```
#> Ibm
```

```
#> 2000-01-01 78.6
```

```
#> 2000-01-03 82.0
```

```
#> 2000-01-04 79.2
```

```
#> 2000-01-05 82.0
```

```
#> 2000-01-06 80.6
```

```
#> 2000-01-07 80.2
```

На перший погляд цей висновок може збити з пантелику. Ми попросили "2 week", а `xts` повернув шість днів. Це може здатися дивним, поки ми не побачимо на календар січня 2000 року.

За календарем видно, що в першому тижні січня 2000 року є тільки один день, субота - 1-е число. Потім йде другий тиждень з 2-ого по 8-е число. У вибраних даних немає значення для 8-го числа, тому, коли використовується функція `first`, щоб запросити перші два тижні, вона повертає всі значення за перші два календарні тижня. У поданому прикладі перші два календарні тижня містять тільки шість значень.

Точно так само можна попросити функцію `last` дати нам дані за останній місяць:

```
last(ibm, "month")
```

```
#> ibm
```

```
#> 2017-12-01 152
#> 2017-12-04 153
#> 2017-12-05 152
#> 2017-12-06 151
#> 2017-12-07 150
#> 2017-12-08 152
#> 2017-12-11 152
#> 2017-12-12 154
#> 2017-12-13 151
#> 2017-12-14 151
#> 2017-12-15 149
#> 2017-12-18 150
#> 2017-12-19 150
#> 2017-12-20 150
#> 2017-12-21 148
#> 2017-12-22 149
#> 2017-12-26 150
#> 2017-12-27 150
#> 2017-12-28 151
#> 2017-12-29 150
```

Якби тут були використані об'єкти zoo, необхідно було б перетворити їх в об'єкти xts, перш ніж передавати їх у функції first і last, оскільки це функції xts.

У пакеті tidyverse dplyr також є функції first і last. Якщо робочий процес включає в себе завантаження пакетів xts і dplyr, необхідно переконатися, що явно вказано, яку функцію викликано, використовуючи нотацію package :: function (наприклад, xts :: first).

**В третьому розділі** розглянуто способи та варіанти моделювання та дослідження отриманих результатів. Пропонується почати з підгонки моделі. Функція auto.arima в пакеті forecast може вибрати правильний порядок моделі і підігнати модель під обрані дані: library(forecast); auto.arima(x).

Якщо порядок моделі  $(p, d, q)$  вже відомий, то функція `arima` може підігнати модель безпосередньо: `arima(x, order = c(p, d, q))`

Створення моделі ARIMA включає в себе три етапи:

- 1) визначення порядку моделі;
- 2) підгонку моделі до даних із зазначенням коефіцієнтів;
- 3) застосування діагностичних заходів для перевірки моделі.

Порядок моделі зазвичай позначається трьома цілими числами  $(p, d, q)$ , де  $p$  - число коефіцієнтів авторегресії,  $d$  - ступінь диференціювання, а  $q$  - число коефіцієнтів змінного середнього.

Висновок, отриманий на даному етапі ілюструє головну проблему, яка виникає при моделюванні ARIMA: не всі коефіцієнти обов'язково є значущими. Якщо один з інтервалів містить нуль, істинний коефіцієнт може бути нульовим сам по собі, і в цьому випадку складова не потрібна.

Якщо виявлено, що модель містить незначні коефіцієнти, рекомендується використовувати метод по їх видаленню. Функції `auto.arima` і `arima` містять корисні властивості для підгонки найбільш відповідної моделі. Наприклад, можна змусити їх включити або виключити компонент тренда.

І останнє зауваження: небезпека функції `auto.arima` полягає в тому, що вона робить моделювання ARIMA простим. Моделювати ARIMA не просто. це більше мистецтво, ніж наука, і автоматично згенерована модель є лише відправної точкою.

Наступним кроком пропонується дослідити виконання діагностики моделі. Функція `arima` включає в себе параметр `fixed`, який є вектором. Вектор повинен містити один елемент для кожного коефіцієнта в моделі, включаючи член дрейфу (якщо такий є). Кожен елемент має значення NA або 0. Для незмінних коефіцієнтів варто використовувати значення NA і 0 для коефіцієнтів, які потрібно видалити. У цьому прикладі показана модель ARIMA (2, 1, 2) з першим авторегресійним коефіцієнтом і першим коефіцієнтом змінного середнього, рівним 0:

`arima(x, order = c(2, 1, 2), fixed = c(0, NA, 0, NA))`

Пакет `fpp2` містить набір даних під назвою `euroretail`, який є квартальним роздрібним індексом зони євро. Давайте виконаємо функцію `auto.arima` для даних і подивимося на 98% довірчі інтервали

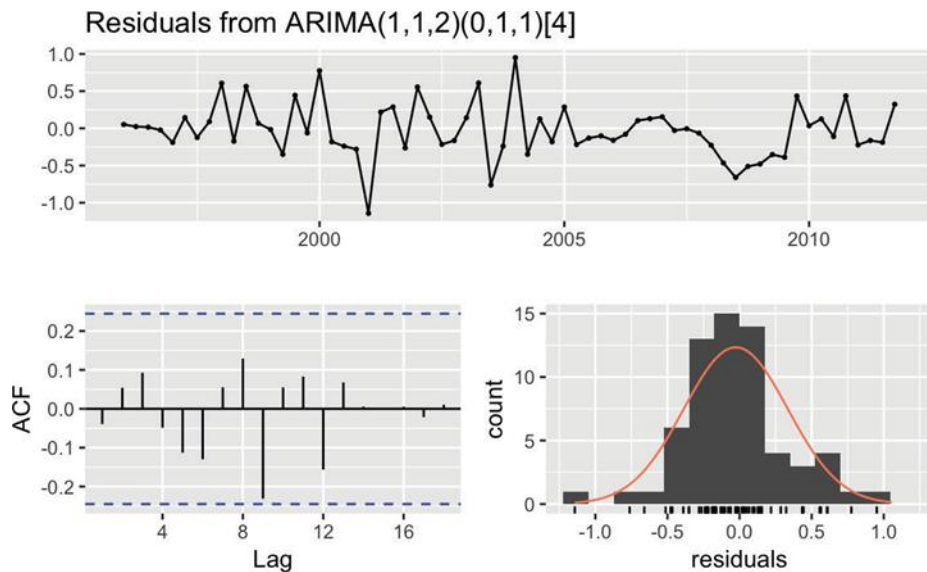


Рис. 1.2. Графік нев'язок, хороша модель

Ось що добре в цих графіках:

- стандартизовані нев'язки не показують кластери волатильності;
- автокореляційна функція (ACF) не вказує значущою автокорреляції між нев'язками;
- нев'язки мають форму дзвони, що говорить про те, що вони досить симетричні;
- $p$ -значення в тесті Льюнг-Боксу велике, а це вказує на те, що нев'язки не мають шаблонів, тобто вся інформація була витягнута моделлю і залишився тільки шум.

Для порівняння, на Рис. 1.3. показані діагностичні діаграми з проблемами:

- автокореляційна функція показує значні автокорреляції між нев'язке;
- $p$ -значення в тесті Льюнг-Боксу невеликі, що вказує на наявність певної закономірності в нев'язці (тобто є інформація, яку необхідно витягти з даних);
- нев'язки виглядають асиметрично.

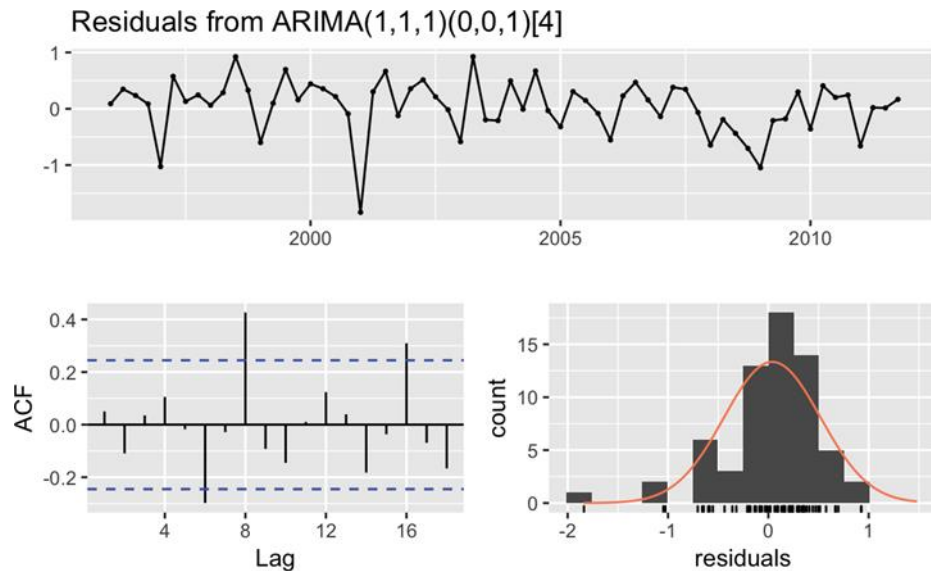


Рис. 1.3. Графік нев'язок, проблемна модель

Це базова діагностика, але початок добрий. Перш ніж зробити висновок, що ARIMA модель справна, необхідно виконати рекомендовані діагностичні тести. Додаткові перевірки нев'язок можуть включати в себе:

- тести на нормальність;
- графік квантиль-квантиль (Q-Q);
- точкову діаграму для підігнаних значень.

Наступним кроком можна провести прогнозування за моделлю ARIMA. Функція `forecast` розрахує наступні кілька спостережень і їх стандартні помилки відповідно до моделі. Вона повертає список з 10 елементів. Коли ми виводимо модель, як продемонстровано вище, функція повертає точки часового ряду, які вона прогнозує, прогноз і дві пари довірчих інтервалів: максимум / мінімум 80% і максимум / мінімум 95%.

Поширеним тестом на наявність повернення до середнього є доповнений тест Дікі-Фуллера (ADF), який реалізується функцією `adf.test` з пакета `tseries`.

Висновок функції `adf.test` включає в себе  $p$ -значення. Традиційно, якщо  $p < 0,05$ , часовий ряд, ймовірно, повертається до середнього значення, тоді як  $p > 0,05$  не дає таких доказів.

Коли тимчасовий ряд повертається до середнього, він прагне повернутися до своєї довгострокової середнього значення. Значення може «гуляти», але врешті-

решт повернеться. Якщо часовий ряд не повертається до середнього, він може зміститися, так і не повернувшись до середнього значення.

Функція `adf.test` ущільнює ваші дані перед виконанням тесту Дікі-Фуллера. Спочатку вона автоматично видаляє тренд з даних, а потім повторно центрує дані, даючи їм середнє значення нуля.

Якщо подібного роду операції небажані для застосування, необхідно використовувати натомість функцію `adfTest` з пакета `fUnitRoots`.

Також в даному розділі розглянуто ряд інших операцій по роботі з часовими рядами:

- підгонка моделі `arima`;
- видалення незначущих коефіцієнтів `ARIMA`;
- виконання діагностики для моделі `ARIMA`;
- прогнозування за моделлю `ARIMA`;
- побудова прогнозу;
- тестування на наявність повернення до середнього;
- згладжування часового ряду.

**У четвертому розділі** наводиться опис розробленого програмно-алгоритмічного забезпечення.

**У спеціальній частині** магістерської кваліфікаційної роботи з «Охорони праці та безпеки у надзвичайних ситуаціях» розглянуто мікрокліматичні умови праці на робочих місцях на предмет виробничого освітлення та дотримання вимог експлуатації ПК. В результаті розрахунків встановлено, що передбаченої кількості вікон розглянутого для прикладу приміщення, тобто їх загальної площі вистачає для забезпечення вимог санітарних норм щодо природного освітлення приміщення для якого проводився аналіз умов праці. Визначено, що для забезпечення штучного освітлення слід використовувати 10 світильників, які необхідно комплектувати 2 люмінесцентними лампами типу ЛДЦ потужністю 30 Вт та довжиною 0,9 м кожна. Загальна потужність освітлення складає 600 Вт. Розроблено інструктаж з техніки безпеки під час надзвичайних ситуацій.

У методичній частині розроблено практичні роботи на теми «Розкладання часового ряду на складові» та «Прогнозування часових рядів з використанням моделей ARIMA».



## ЗАГАЛЬНІ ВИСНОВКИ

У даній магістерській кваліфікаційній роботі досліджено методи прогнозування на основі моделі ARIMA. У першому розділі проведено аналіз предметної області, досліджено роль і місце прогнозування в сфері фінансів та бізнесу. Проведений аналіз публікацій дав змогу зрозуміти, що перспективним напрямом у вирішенні задачі прогнозування та подальшого вибору основних напрямків розвитку є прогнозування на основі моделі ARIMA з дослідженням історичних даних поданих засобами перетворення у форму часових рядів. У другому розділі розглянута мова R як засіб ілюстрації роботи з часовими рядами, проведено дослідження способів побудови та представлення часових рядів, також перераховані та продемонстровані приклади типових операції над часовими рядами, таких як: витяг найстаріших або найостанніших спостережень, вибір елементів з тимчасового ряду, об'єднання декількох часових рядів, заповнення тимчасового, зсув тимчасового ряду, обчислення послідовних, виконання розрахунків по часових рядах, обчислення ковзної середньої, застосування функції по календарному періоду, застосування функції `rollapply`, побудова функції автокореляції, тестування тимчасового ряду на наявність автокореляцій, побудова функції часткової автокореляції, пошук кореляцій з тимчасовим лагом між двома часовими рядами. В доповнення до вище представленої інформації, у третьому розділі прокоментована реалізація підгонки моделі `arima`, видалення незначущих коефіцієнтів `arima`, виконання діагностики для моделі `arima`, прогнозування за моделлю `arima`, побудова прогнозу, тестування на наявність повернення до середнього та згладжування часового ряду. Розроблено систему прогнозування фінансових показників на основі моделі ARIMA.

У методичній частині магістерської роботи розроблено дві практичні роботи на теми «Розкладання часового ряду на складові» та «Прогнозування часових рядів з використанням моделей ARIMA».

У спеціальній частині магістерської роботи з «Охорони праці та безпеки у надзвичайних ситуаціях» розглянуто мікрокліматичні умови праці на робочих місцях на предмет виробничого освітлення та дотримання вимог експлуатації ПК.

Виконано перевірочний розрахунок природного освітлення та розраховано загальне рівномірне освітлення люмінесцентними лампами в розглянутому приміщенні. Розроблено інструктаж для дій працівників та керівництва компанії на випадок виникнення типових надзвичайних ситуацій, а саме: при пожежі, землетрусі, урагані, грозі.

## АНОТАЦІЯ

### **Танзі Сана. Інформаційна система прогнозування фінансових показників на основі моделі ARIMA. - На правах рукопису**

Магістерська кваліфікаційна робота на здобуття освітньої кваліфікації «Магістр з системного аналізу». – Чорноморський національний університет імені Петра Могили, Миколаїв, 2021.

У сучасному світі програмне забезпечення нероздільно пов'язане з повсякденним життям будь якої людини. Одними з найчастіше використовуваними програмними додатками є, безумовно, банківські програмні системи, які дозволяють користувачеві керувати своїми фінансами безпосередньо зі свого комп'ютеру. Але з розповсюдженням фінансових програм також з'являється питання їх безпеки та коректної роботи та забезпечення стабільності фінансових установ. Таким чином будь яка фінансова установа особливо на старті, або в часи нестабільності потребує числового обґрунтування рентабельності своєї діяльності. Саме це і стало основною метою та завдання даної роботи: дослідження та імплементація інформаційної системи, яка спрогнозує заплановану активність на основі попереднього досвіду.

Пояснювальна записка магістерської роботи складається зі вступу, трьох розділів, висновків, переліку джерел посилання, 2 додатків та спеціальної частини з охорони праці.

У вступі визначається актуальність теми, наведені задачі, які заплановано вирішити для досягнення поставленої мети. У першому розділі розглядаються існуючі алгоритми, які дозволяють здійснити аналіз часових рядів. У другому розділі розглянуто математичні моделі, методи, інформаційні технології, які були виявлені та в ході аналізу обрані у якості оптимальної для вирішення поставленої задачі. У третьому розділі описаний процес моделювання, прорахунок та дослідження отриманих результатів. Розділ 4 описує програмну реалізацію та особливості мови, яка була обрана у якості засоба розробки раніше описаного та дослідженого алгоритму.

Робота містить \_\_ рисунків, \_\_ літературних джерела та \_\_ додатків.  
Загальний обсяг дипломної роботи складає сторінок.

***Ключові слова: прогнозування, ARIMA, ACF, фінансування, економіка, рентабельність, Python, R .***

## **ABSTRACT**

### **Tanzy Sanae. Information system for forecasting financial indicators based on the ARIMA model. - On the rights of the manuscript**

Master's qualification work for obtaining the educational qualification "Master of Systems Analysis". - Petro Mohyla Black Sea National University, Mykolaiv, 2021.

In a globalized world, software is inextricably linked to everyone's routine. One of the most frequently used software applications is, of course, banking software systems, which allow users to manage their finance directly from their computers. But with the widespread of financial programs, there also appears the question of their security and proper operation and the stability of financial institutions. Thus, every financial institution, especially at the beginning or in times of instability, needs particular numerical justification of the profitability of its activities. This is the main purpose and object of this work: research and implementation of an information system that predicts the planned activity based on previous experience.

The explanatory note of the master's thesis consists of an introduction, three sections, conclusions, a list of reference sources, 2 appendices and a special section on labor protection.

The introduction determines the relevance of the topic, the tasks that are planned to be solved to achieve this goal. The first section discusses the existing algorithms that allow the analysis of time series. The second section considers mathematical models, methods, information technologies that have been identified and selected during the analysis as optimal for solving the problem. The third section describes the process of modeling, calculation and study of the results. Section 4 describes the software implementation and features of the language, which was chosen as a means of developing the previously described and researched algorithm.

The work contains \_\_ drawings, \_\_ literature sources and \_\_ appendices. The total volume of the thesis is pages.

*Key words: forecasting, ARIMA, ACF, financing, economy, profitability, Python, R.*