Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра математики факультету інформатики

Курсова робота

за спеціальністю 113 Прикладна математика

освітня програма «Прикладна математика»

ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ ДЛЯ ПРОГНОЗУВАННЯ В ТРЕЙДИНГУ

Керівник курсової роботи
к-т фізико-математичних наук, доцент
Дрінь С.С.
_____
(підпис)
“____” _____ 2021 р.

Виконала студентка факультету інформатики
спеціальності «Прикладна математика» - 3 курс
Шульга В.К.
“____” _____ 2021 р.

Київ 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра математики факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри математики,

проф., д.ф.-м.н.

_____ Б. В. Олійник

(підпис)

«____»_____2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студентці 3-го курсу факультету інформатики

Шульзі Вірі Костянтинівні

Тема: Побудова математичної моделі для прогнозування в трейдингу

Вихідні дані: Досліджено метод TRAMO для побудови ARIMA моделі

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Календарний план

Анотація

Вступ

1. ARIMA модель, її складові і властивості
2. Основні принципи роботи методу TRAMO/SEATS і його застосування. Порівняння результатів з X-13

Висновки

Перелік прийнятих скорочень

Використана література

Дата видачі „___” _____ 2021 р. Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Тема: Побудова математичної моделі для прогнозування в трейдингу

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Визначення теми курсової роботи.	29.10.2020	
2.	Визначення літератури для роботи.	16.03.2021	
3.	Огляд і аналіз літератури за темою роботи.	20.03.2021	
4.	Оформлення теоретичної частини курсової роботи.	31.03.2021	
5.	Виконання практичного застосування вибраного методу.	31.04.2021	
6.	Оформлення практичної частини курсової роботи.	12.05.2021	
7.	Створення презентації курсової роботи.	18.05.2021	
8.	Захист курсової роботи.	20.05.2021	

ЗМІСТ

АНОТАЦІЯ

У даній роботі розглянуто модель ARIMA і алгоритм, який будується на її основі, — TRAMO/SEATS. Описано основні кроки побудови моделі за допомогою TRAMO, як-от: можлива трансформація даних, робота з пропущеними значеннями, ідентифікація моделі, пошук її параметрів порядків. Наведено приклад застосування TRAMO для ціни на акцію Nvidia, із заданими параметрами. Проведено аналіз отриманих результатів. Порівнюється якість прогнозу з алгоритмом X-13.

ВСТУП

Прогнозувати ціну в трейдингу можна за допомогою різних моделей: багатофакторна (лінійна і нелінійна) регресія, різні різновиди авторегресійних моделей (наприклад, AR, MA, ARMA, ARIMA), певні методи (наприклад, парний терйдинг). Проблемою прогнозування є не точно отримані значення. У випадку розглянутих надалі ARIMA моделей є певні фактори, які впливають на точність значень: відсутність деяких спостережень, нелінійність і відсутність стаціонарності часового ряду, значний вплив викидів, неправильне визначення параметрів і порядків моделі, ефекти від робочого дня і свят.

Через раніше зазначені впливи на якість прогнозів моделі, було створено різні методи побудови ARIMA і її сезонного коригування. До другої половини XX століття широко використовувався алгоритм прогнозування з сезонним коригуванням – X-11. Його було розроблено статистичним бюро США. Основні принципи його роботи описали Шискін та Айзенпрес у 1957 році. Він базується на поєднанні статистичних методів: ковзних середніх, обробки нетипових спостережень та коригування трейдиногового дня. У 1980 році в Канаді було створено алгоритм X-11-ARIMA, який для прогнозу і декомпозиції використовував саме ARIMA моделі. Основні принципи його роботи описав Дагум. Надалі статистичним бюро США було введено алгоритми, в яких було покращено роботу X-11-ARIMA: X-12-ARIMA, X-13ARIMA-SEATS. У 1994 році для банку Іспанії було знайдено новий метод побудови ARIMA моделі з сезонним коригуванням – TRAMO/SEATS. Його принцип роботи описали Гомез і Маравал у своїх роботах у 1994 і 2001 роках. Цей метод використовує більш потужну процедуру вибору моделі, яка називається TRAMO. [7]

Метою даної роботи є:

1. Розглянути поняття стаціонарності та його вплив на прогноз. Спосіб перетворення нестаціонарного часового ряду у стаціонарний завдяки обчислення різниці між послідовними спостереженнями.
2. Визначити з яких частин складається ARIMA модель і деякі властивості параметрів її порядків.
3. Описати певні процеси побудови ARIMA моделі за допомогою методу TRAMO.
4. Розглянути метод побудови TRAMO на конкретному прикладі з визначеними параметрами за допомогою програмного забезпечення JDemetra+.
5. Порівняти результат роботи з методом X-13.

1 ARIMA модель, її складові і властивості

1.1 Визначення необхідних інструментів для розуміння AR, MA, ARMA, ARIMA моделей

1.1.1 Стаціонарний процес

Стаціонарний процес – це деякий клас стохастичних процесів, який базується на припущенні, що процес перебуває в певному стані статистичної рівноваги. Стохастичний процес називається суворо стаціонарним, якщо на його властивості не впливає зміна часу, тобто якщо спільний розподіл ймовірностей, пов'язаний із m спостереженнями $z_{t_1}, z_{t_2}, \dots, z_{t_m}$, зробленими в $t_1, t_2, \dots, t_m$ моменти часу, є таким самим, як і для спостережень $z_{t_1+k}, z_{t_2+k}, \dots, z_{t_m+k}$, які виконані в моменти $t_1 + k, t_2 + k, \dots, t_m + k$. Таким чином, щоб процес був стаціонарним, на розподіл будь-якого набору спостережень не повинно впливати зміщення часу на ціле число k. Тобто процеси, які мають тренд і сезонність не є стаціонарними. Але циклічні (які не мають тренду та сезонності) процеси є стаціонарними, через те, що вони не мають фіксованого періоду. У загальному випадку стаціонарний процес не має передбачуваних закономірностей. Також треба зазначити, що стаціонарний процес має постійну дисперсію. [1]

1.1.2 Обчислення різниці, як метод перетворення нестаціонарних даних у стаціонарні

Не завжди дані на початку є у вигляді стаціонарного процесу. Одним із методів перетворення нестаціонарних часових рядів у стаціонарні є обчислення різниці між послідовними спостереженнями $\Delta y_t = y_t - y_{t-1}$. Цей метод допомагає стабілізувати середнє значення часового ряду, усуваючи (зменшуючи) тенденцію і сезонність. [2] Розглянемо це на прикладі ціни на акції Nvidia.

```
import yfinance as yf
import matplotlib.pyplot as plt

data_nvidia = yf.download('NVDA', start='2020-01-01', end='2021-01-01')
data_nvidia = (data_nvidia['Close'] + data_nvidia['Open'])/2
data_nvidia.plot()
plt.ylabel('Price')
plt.show()
```

```
[**********100%**********]  1 of 1 completed
```

Рис. 1.1.1

Вже за графіком з рис 1.1.1 можна побачити, що ціни на акції Nvidia мають зростаючий тренд. Використаємо обчислення однієї різниці між послідовними спостереженнями і матимемо результат, який зображено на рис 1.1.2.

```
import yfinance as yf
import matplotlib.pyplot as plt

data_nvidia = yf.download('NVDA', start='2020-01-01', end='2021-01-01')
data_nvidia = (data_nvidia['Close'] + data_nvidia['Open'])/2
diff_data_nvidia = data_nvidia.diff()
diff_data_nvidia.plot()
plt.ylabel('Difference between price')
plt.show()
```

```
[**********100%**********]  1 of 1 completed
```

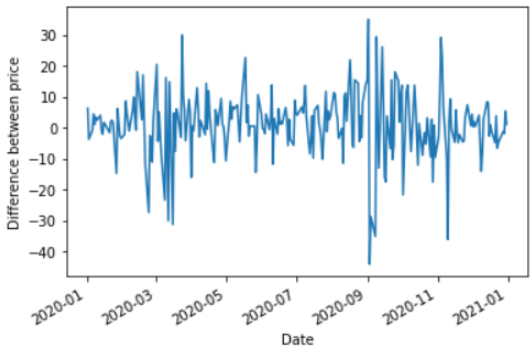


Рис. 1.1.2

З графіка, який зображено на рис 1.1.2, видно, що ми позбавились явного тренду, але це ще не означає, що він є стаціонарним.

1.1.3 Автоковаріація і автокореляція

Припущення про стаціонарність передбачає, що спільний розподіл ймовірностей $p(z_{t_1}, z_{t_2})$ є однаковим для будь-яких моментів часу t_1, t_2, при чому різниця між цими моментами часу є фіксованою. З цього випливає, що коваріація між значеннями спостережень z_t і z_{t+k} [1] (де зрушення в часі k називається часовим лагом і позначається lag k [8]) повинна бути однаковою в будь-який момент часу t за виконання припущення про стаціонарність. Така коваріація називається автоковаріацією за lag k і визначається наступною формулою:

$$\gamma_k = cov[z_t, z_{t+k}] = E[(z_t - \mu)(z_{t+k} - \mu)]$$

Автокореляція за lag k визначається за наступною формулою:

$$\rho_k = \frac{E[(z_t - \mu)(z_{t+k} - \mu)]}{\sqrt{E[(z_t - \mu)^2]E[(z_{t+k} - \mu)^2]}} = \frac{E[(z_t - \mu)(z_{t+k} - \mu)]}{\sigma_z^2}$$

Враховуючи, що ми маємо стаціонарний процес, дисперсія $\sigma_z^2 = \gamma_0$ є такою ж в момент часу $t + k$ як і в момент часу t. Таким чином автокореляція за lag k є кореляцією між z_{t_1}та z_{t_2}і має вигляд:

$$\rho_k = \frac{\gamma_k}{\gamma_0}$$

[1]

Для знаходження автоковаріації для різних значень lag k можна використати функцію plot_acf з бібліотеки statsmodels у Python. Наприклад, на рис 1.1.3 зображено графік з застосуванням автоковаріаційної функції для ціни на акції Nvidia, до яких ми застосували обчислення різниці між послідовними спостереженнями раніше на рис. 1.1.2.

```
import yfinance as yf
import matplotlib.pyplot as plt
import statsmodels.api as sm

data_nvidia = yf.download('NVDA', start='2020-01-01', end='2021-01-01')
data_nvidia = (data_nvidia['Close'] + data_nvidia['Open'])/2
diff_data_nvidia = data_nvidia.diff().dropna()
sm.graphics.tsa.plot_acf(diff_data_nvidia, lags=40, zero=False)
plt.xlabel('Lag')
plt.ylabel('ACF value')
plt.show()
```

```
[**********100%**********]  1 of 1 completed
```

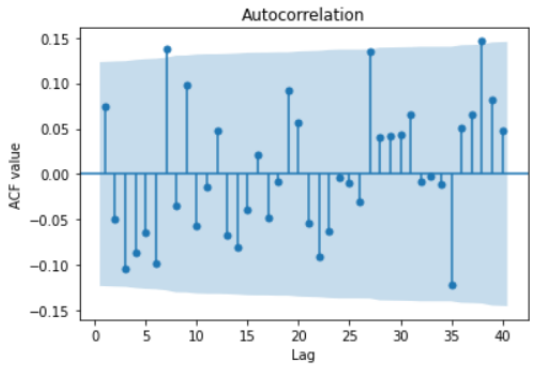


Рис. 1.1.3

1.2 AR, MA, ARMA, ARIMA моделі

1.2.1 Означення AR, MA, ARMA, ARIMA моделей

Модель авторегресії (autoregressive, AR) – це стохастична модель, в якої поточне значення процесу виражається як скінчена, лінійна сукупність попередніх значень процесу та випадкової помилки. Припустимо, що в нас відомо значення попередніх $t, t-1, t-2, \ldots$ спостережень $z_1, z_{t-1}, z_{t-2}, \ldots$, які зроблено через однакові проміжки часу. Нехай $\tilde{z}_t = z_t - \mu$ буде прогнозованим часовим рядом із відхиленням μ від дійсних значень. Тоді маємо, що AR(p) має наступний загальний вигляд:

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + \cdots + \phi_p \tilde{z}_{t-p} + a_t \qquad (1.2.1)$$

де ϕ_p – коефіцієнти, які необхідно оцінити, $\tilde{z}_{t-1}, \ldots, \tilde{z}_{t-p}$ – попередні p спостережень, a_t – поточна похибка, яка є білим шумом. [1] Білий шум – це часовий ряд, в якого відсутня автокореляція (тобто значення автокореляційної функції при будь-якому $lag\ k$ наближається до нуля). [2]

Модель ковзного середнього (moving-average, MA) – це лінійна модель, в якої значення процесу виражається за допомогою скінченної кількості q минулих випадкових помилок та нової випадкової помилки. Зробимо

припущення щодо часового ряду $\tilde{z}_t$ таке саме, як і для авторегресійної моделі (AR). Тоді маємо, що MA(q) має наступний загальний вигляд:

$$\tilde{z}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q} \quad (1.2.2)$$

де θ_q – це коефіцієнти, які необхідно оцінити, $a_{t-1}, \ldots, a_{t-q}$ – q попередніх похибок, a_t – поточна похибка, яка є білим шумом.

Модель авторегресії - ковзного середнього (autoregressive moving average, ARMA) – це модель, яка є змішуванням моделей авторегресії і ковзного середнього, тобто в якої поточне значення процесу виражається за допомогою скінченої, лінійної сукупності попередніх значень процесу й минулих випадкових помилок. Підсумувавши загальний вигляд AR(p) і MA(q) відповідно у формулах (1.2.1) і (1.2.2), маємо загальний вигляд ARMA(p, q):

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \cdots + \phi_p \tilde{z}_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q} \quad (1.2.3)$$

Несезонна інтегрована модель авторегресії – ковзного середнього – це модель ARMA, до якої застосовано обчислення різниці між послідовними спостереженнями d разів. Загальний вигляд ARMA було визначено у (1.2.3). Таким чином маємо загальний вигляд ARIMA(p, d, q):

$$w_t = \phi_1 w_{t-1} + \cdots + \phi_p w_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}$$

де $w_t = \Delta^d z_t$ (обчислення різниці між поточними спостереженнями d разів), ϕ_p – коефіцієнти, які треба оцінити, $w_{t-1}, \ldots, w_{t-p}$ – p попередніх значень d різниць спостережень, a_t – поточна похибка, яка є білим шумом, $a_{t-1}, \ldots, a_{t-q}$ – q попередніх похибок, θ_q – коефіцієнти, які необхідно оцінити
[1]

1.2.2 Лаговий оператор. Вигляд AR, MA, ARMA моделей з використанням лагового оператора

У моделях, які були раніше розглянуті, ми можемо виразити поточне значення спостереження, а для того, щоб виразити значення спостереження у момент часу t-1 за допомогою спостереження в момент часу t, використовують лаговий оператор. Таким чином маємо:

$$By_t = y_{t-1} \quad (1.2.4)$$

$$y_{t-2} = By_{t-1} = B(By_t) = B^2 y_t \quad (1.2.5)$$

З (1.2.4) і (1.2.5) безпосередньо випливає:

$$y_{t-k} = B^k y_t \quad (1.2.6)$$

[2]

Тоді авторегресійний процес AR(p) можна записати за допомогою лагового оператору. Використавши (1.2.6) підставимо у (1.2.1):

$$\tilde{z}_t = \phi_1 B\tilde{z}_t + \phi_2 B^2\tilde{z}_t + \cdots + \phi_p B^p\tilde{z}_t + a_t$$

Далі поділимо на $\tilde{z}_t$ праву і ліву частину. Матимемо:

$$1 = \phi_1 B + \phi_2 B^2 + \cdots + \phi_p B^p + \frac{a_t}{\tilde{z}_t}$$

$$1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p = \frac{a_t}{\tilde{z}_t}$$

$$\left(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p\right)\tilde{z}_t = a_t$$

Позначимо поліном $1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ виразом $\phi(B)$.

Отримаємо:

$$\phi(B)\tilde{z}_t = a_t \quad (1.2.7)$$

Також процес ковзного середнього MA(q) можна записати за допомогою лагового оператору. Використавши (1.2.6) у (1.2.2) маємо наступний вигляд:

$$\tilde{z}_t = a_t - \theta_1 B a_t - \theta_2 B^2 a_t - \cdots - \theta_q B^q a_t$$

$$\tilde{z}_t = a_t(1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q)$$

Позначимо поліном $1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q$ виразом $\theta(B)$.

Отримаємо:

$$\tilde{z}_t = \theta(B)a_t \quad (1.2.8)$$

Підсумовуючи (1.2.7) і (1.2.8) маємо, що ARMA(p, q) модель записана за допомогою лагового оператору буде виглядати наступним чином:

$$\phi(B)\tilde{z}_t = \theta(B)a_t$$

де $\phi(B)$, $\theta(B)$ є поліномами оператору B, відповідно степеней p і q.

1.2.3 Корені авторегресійного поліному в ARMA моделі

Для того, щоб ARMA процес був стаціонарним необхідною умовою є те, що корені $\phi(B) = 0$ повинні бути за модулем більші одиниці. Природнім чином процес стає нестаціонарним, якщо не виконується ця умова. Це легко побачити на прикладі. Візьмемо ARMA модель першого порядку:

$$(1 - \phi B)\tilde{z}_t = a_t$$

Коли $|\phi| < 1$, модель є стаціонарною. Покладемо $\phi = 3$ – значення за якого модель не буде стаціонарною. Нехай a_t – це набір одиничних випадкових стандартних відхилень ($\sigma_a^2 = 1$). Зафіксуємо $\tilde{z}_0 = 0{,}2$ Матимемо модель:

$$\tilde{z}_t = 3\tilde{z}_{t-1} + a_t$$

Побудуємо її графік, використавши бібліотеку matplotlib для зображення графіку і normal з numpy для значення a_t.

```
import matplotlib.pyplot as plt
from numpy.random import normal

z_0 = 0.2
fi = 2
z_i = []
z_i.append(fi*z_0 + normal())
for i in range(8):
    z_i.append(fi*z_i[-1])
plt.figure(figsize=(5,5))
plt.scatter(range(1, 10), z_i)
plt.xlabel('Number of observation')
plt.ylabel('Observation value')
plt.show()
```

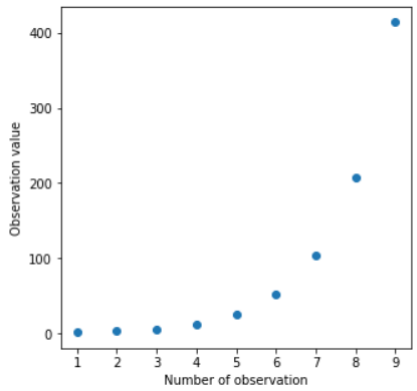


Рис. 1.2.4

З графіку на рис. 1.2.4 можна побачити, як за короткий період авторегресійний процес починає експоненційно зростати, причому випадкове значення a_t майже ніяк не впливає на поточне значення. Поведінка нестаціонарних часових рядів, породжених процесами вищих порядків, є схожою. Крім того, ця поведінка по суті є однаковою, незалежно від того, чи введено змінні процесу ковзного середнього.

Отже, ми розглянули випадки, коли корінь $\phi(B) = 0$ є строго більшим одиниці і строго меншим одиниці. Залишилось розглянути, що відбудеться, коли корінь дорівнює одиниці. Такі моделі мають цінність через інтерпретацію однорідної нестаціонарності часових рядів. Розглянемо модель вигляду:

$$\varphi(B)\tilde{z}_t = \theta(B)a_t$$

де $\varphi(B)$ – нестаціонарний авторегресійний оператор, такий що d його коренів $\varphi(B) = 0$ дорівнюють одиниці, а модуль інших строго більше за одиницю. Тоді можна записати модель у вигляді:

$$\varphi(B)\tilde{z}_t = \phi(B)(1-B)^d\tilde{z}_t = \theta(B)a_t$$

де $\phi(B)$ – саціонарний автегресійний оператор. Оскільки $\Delta^d\tilde{z}_t = \Delta^d z_t$ для $d \geq 1$, де $\Delta = 1 - B$ – оператор різниці, ми можемо записати модель у вигляді:

$$\phi(B)\Delta^d z_t = \theta(B)a_t \qquad (1.2.9)$$

Можемо визначити еквівалентно наступними двома рівностями:

1. $$\phi(B)w_t = \theta(B)a_t$$
2. $$w_t = \Delta^d z_t$$

Таким чином можемо побачити, що d-а різниця часового ряду може бути представлена стаціонарним, оборотним ARMA процесом. Отже, формула вигляду (1.2.9) називається інтегрованим процесом авторегресійного-ковзного середнього (ARIMA). [1]

1.2.4 Загальна форма ARIMA моделі

Розглянемо деяке розширення ARIMA, в якому додали деяку константу θ_0, що дає більш загальну форму:

$$\varphi(B)z_t = \phi(B)\Delta^d z_t = \theta_0 + \theta(B)a_t \qquad (1.2.10)$$

де

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q$$

Маємо, шо $\phi(B)$ будемо називати авторегресійним оператором, який вважається стаціонарним, що означає, що його корені за модулем більше одиниці.

Також отримаємо, що $\varphi(B)z_t = \phi(B)\Delta^d z_t$ будемо називати узагальненим авторегресійним оператором, тобто це нестаціонарний оператор з d коренями $\varphi(B) = 0$, які дорівнюють одиниці.

Будемо називати $\theta(B)$ оператором ковзного середнього, враховувавши, що він є оборотним, маємо, що корені $\theta(B) = 0$ за модулем більше одиниці.

Коли d = 0 модель зображує стаціонарний процес. Вимоги до стаціонарності застосовуються незалежно один від одного, бо в загальному випадку оператори ковзного середнього й авторегресії не будуть мати однаковий порядок.

Якщо прибрати константу θ_0, тоді модель вигляду (1.2.10) показує часовий ряд із стохастичним трендом, що характеризується, наприклад, випадковими змінами рівня й нахилу часового ряду. [1]

1.2.5 Сезонна ARIMA модель

Форма ARIMA моделі, яка згадана раніше, є несезонною, тобто вона не включає сезонну компоненту. Сезонна ARIMA модель формується шляхом включення додаткових сезонних членів у модель ARIMA, тобто сезонна ARIMA модель має вигляд:

$$ARIMA\ (p, d, q)\ (P, D, Q)_s$$

де P – це порядок сезонної авторегресії, D – це порядок сезонної різниці, Q – це порядок сезонного ковзного середнього, s – кількість спостережень за рік. [2]
Загальна форма сезонної ARIMA наступна:

$$\phi_p(B)\Phi_P(B^s)\Delta^d\Delta_s^D z_t = \theta_q(B)\Theta_Q(B^s)a_t$$

де $\phi_p(B)$ – це несезонний авторегресійний поліном степеню p, $\Phi_P(B^s)$ – це сезонний авторегресійний поліном степеню P, Δ^d – несезонна різниця степеню d, Δ_s^D – сезонна різниця степеню D, z_t – поточне спостереження, $\theta_q(B)$ – поліном несезонного ковзного середнього, $\Theta_Q(B^s)$ – це поліном сезонного ковзного середнього, a_t – поточка похибка, яка є білим шумом.

[1]

1.2.6 Декомпозиція і сезонне коригування

Дані часових рядів часто можуть відображати різні типи. Якщо дані розподілити на декілька компонент, кожна з яких буде відображати певний шаблон, може бути дуже корисним для аналізу часових рядів і прогнозування. Часто дані розділяють на циклічно-трендову, сезонну і залишкову (неконтрольовану) компоненту. Деякими із способів вираження поділу на компоненти є сума, добуток, сума логарифмів.

У сезонно скоригованих даних видалена саме компонента сезонності, завдяки чому ми можемо побачити лише зміну циклу, тренду і залишкову компоненту. Це може бути корисно, коли ми розглядаємо, наприклад, рівень безробіття. Зростання рівня безробіття через те, що випускники шукають роботу, має сезонний характер. Проте зростання рівню безробіття через те, що в країні економічний спад, не має сезонного характеру. Тому більшість аналітиків, які досліджують дані на рівень безробіття, зацікавлені саме в зміні сезонно скоригованих даних. [2]

2 Основні принципи роботи методу TRAMO/SEATS і його застосування. Порівняння результатів з X-13

2.1 Метод побудови ARIMA моделі і її декомпозиції за допомогою TRAMO/SEATS

1.2.7 Загальне означення методу TRAMO/SEATS

TRAMO/SEATS (Time Series Regression with Arima Noise, Missing Observations, and Outliers/Signal Extraction in ARIMA Time Series) – це регресійна модель часових рядів ARIMA з шумом, пропущеними спостереженнями і викидами/ вилучення сигналу з моделей ARIMA часових рядів. Загалом TRAMO будує регресійну модель ARIMA, а SEATS робить декомпозицію. Початкову версію цього методу розробили в 1994 році Агустін Маравал і Віктор Гомез для іспанського банку. Надалі буде розглянуто лише TRAMO.

Метод TRAMO побудови регресійної ARIMA моделі включає наступне:

- пошук викидів (outliers)
- трансформація (перетворення) даних

- інтерполяція пропущених спостережень
- ідентифікація моделі
- оцінка параметрів моделі

1.2.8 Викиди (outliers) в TRAMO.

Виявлення і корекція викидів здійснюється за процедурою, подібною до алгоритму, який навели Чень і Лю (1993).

Процедура автоматичного виявлення викидів. Спочатку обраховуються регресійні параметри (якщо маємо regARIMA) за допомогою методу найменших квадратів і далі параметри ARMA моделі оцінюються з двох регресій, як у Ханнана і Ріссанена (1982). [3]

Процедура Ханнана і Ріссанена (1982) полягає в тому, що порядки авторегресії і ковзного середнього оцінюються за допомогою ряду регресій $y_{t-1}, \dots, y_{t-p}, a_{t-1}, \dots, a_{t-q}$, де y_t- це значення спостереження в час t, a_t – похибка у час t. a_tотримуються шляхом застосування процесу авторегресії до даних, причому використовується те, що послідовність регресій при p = q може бути рекурсивно економічно розрахована шляхом вбудовування їх у послідовність із цих двох регресій. [4]

Далі за допомогою фільтру Калмана [9] знаходимо залишки ряду і отримаємо нові оцінки параметрів регресії. Для кожного спостереження обчислюються t-тести для чотирьох типів викидів: інноваційні викиди (IO, innovational outliers), адитивні викиди (AO, additive outliers), зсув рівня (LS, level shift) і тимчасова зміна (TC, transient change). Далі на рис. 2.1.1, рис. 2.1.2, рис. 2.1.3, рис. 2.1.4 зображено червоною лінією, як саме цей викид виглядає порівняно з загальним часом рядом, який зображено синьою лінією. Кожен тип викиду прибирається один за одним і оцінюються нові параметри для моделі. Після завершення цього етапу, виконується багатофакторна регресія, і, якщо присутні деякі викиди, тоді програма повертається до першої послідовності і повторюється до тих пір, доки не буде усунуто всі викиди для багатофакторної регресії. Особливістю цього алгоритму є те, що всі розрахунки базуються на методах лінійної регресії, що зменшує обчислювальний час. [3]

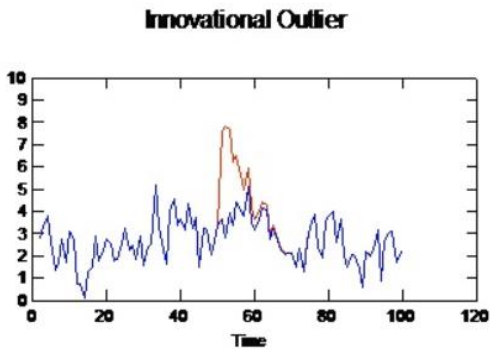


Рис. 2.1.1

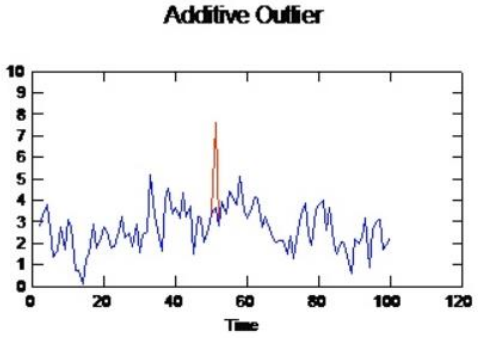


Рис.2.1.2

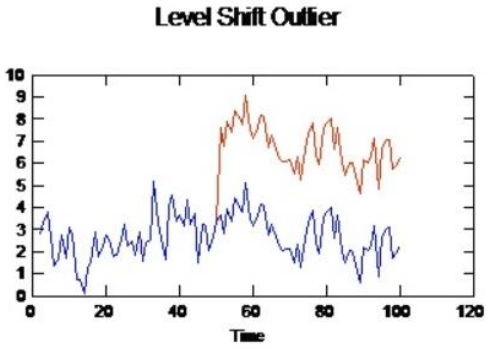


Рис. 2.1.3

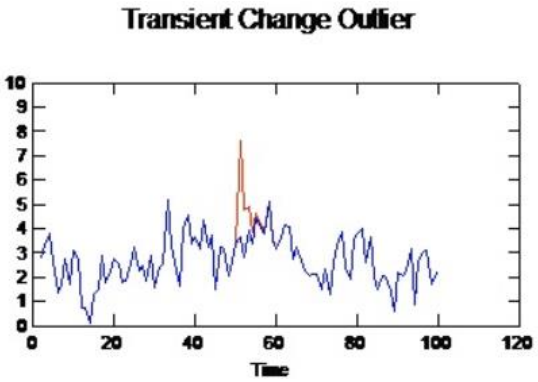


Рис.2.1.4

[10]

Тепер згадаємо певні особливості, які з'явилися у новій версії TRAMO/SEATS у 2014 році. Сезонні викиди можуть бути додані за допомогою автоматичного виявлення викидів. Сезонним викидом називається зміна рівня, яка діє лише в певний період. AIO – параметр, який визначає викиди. DELTATC - параметр демпфування (процес гасіння коливань) тимчасових викидів, за замовченням дорівнює 0.7.

VA – параметр критичного рівня за замовченням, який встановлено для виявлення значущих викидів. Якщо перший хід автоматичної ідентифікації моделі дає залишки, які мають велику автокореляцію, тоді значення VA зменшуються у кількість разів, яке визначає параметр PC (за замовчуванням дорівнює 0.1). [5]

1.2.9 Трансформація (перетворення) даних у TRAMO.

Ще в початковій версії TRAMO є параметр LAM, за допомогою якого користувач може зробити логарифмування даних вказавши значення рівним 1 (за замовчування воно дорівнює 0, тобто логарифмування не відбувається). [3]

Але в останній версії TRAMO за 2014 рік виконується вибір типу перетворення, який визначається автоматично за допомогою log/level тестів. [5] Цей тест базується на оцінці за допомогою функції максимальної правдоподібності параметру λ з перетворення Бокса-Кокса (яке є степеневим перетворенням таким, що значення перетвореного часового ряду є монотонною функцією спостережень), тобто:

$$y_i^{\alpha\prime} = \begin{cases} \dfrac{(y_i^{\alpha} - 1)}{\lambda}, & \lambda \neq 0 \\ \log y_i^{\alpha}, & \lambda = 0 \end{cases}$$

Автоматична процедура спочатку застосовує модель $ARIMA(0,1,1)(0,1,1)_s$ на даних до яких застосовано степеневу зміну ($\lambda = 1$) та на даних, до яких застосовано логарифмування ($\lambda = 0$). Далі порівнюється сума квадратів помилок моделі, яку визначено на даних із степеневою зміною (позначимо SSE_1) і сума квадратів помилок, помножену на середнє геометричне, моделі, яку визначено на логарифмованих даних (позначимо SSE_2). Використовується логарифмування, якщо значення SSE_2 є більшим за значення SSE_1. Також у TRAMO є параметр FCT, за допомогою якого перевіряється зміщення в попередньому log/level тесті. Відповідно якщо FCT > 1, тоді віддається перевага степеневому перетворенню, а якщо FCT < 1, тоді віддається перевага логарифмуванню. [6]

1.2.10 Пропущені спостереження

Встановлювання відсутніх спостережень може відбуватися двома способами. У початковій версії TRAMO є параметр INTERP, за допомогою якого можна визначити спосіб встановлювання пропущених спостережень (за замовчування дорівнює 0, що означає, що немає інтерполяції неспостережуваних значень). Першим є підхід пропуску (INTERP=1), який оцінює за допомогою методу максимальної правдоподібності параметри моделі і використовує деякий алгоритм згладжування. Другим є підхід адитивних викидів (INTERP=2), який полягає в тому, що замість пропущених значень встановлюються випадкові значення, а потім проводиться оцінка за допомогою методу максимальної правдоподібності моделі ARIMA з адитивними

викидами. Якщо використано процедуру автоматичної ідентифікації, тоді пропущені значення обраховуються за допомогою підходу адитивних викидів (INERP = 2). [3]

1.2.11 Процедура автоматичної ідентифікації моделі (AMI)

Алгоритм процедури автоматичної ідентифікації моделі (діаграму даного алгоритму наведено на рис. 2.1.5).

1. Чи є пропущені значення спостережень?

   Якщо так, тоді переходимо до п. 1.1.

   Якщо ні, тоді визначаємо першу $ARIMA(p,d,q)(P,D,Q)_s$ і переходимо до п.2.

   1.1. Проводимо тест на сезонність overSeasTest1 на початкових даних (перед цим до даних може бути застосовано логарифмування, якщо LAM = 1).

   Якщо є сезонність, тоді визначаємо першу $ARIMA(0,1,1)(0,1,1)_s$ (Airline model) і переходимо, до п.2.

   Якщо немає сезонності, тоді визначаємо першу ARIMA, як IMA(1, 1) і переходимо до п.2.

2. Проводимо тест на сезонність overSeasTest2 на логарифмованих даних.

   Якщо сезонності в лінеаризованих даних за тестом overSeasTest2 не знайдено й один з сезонних параметрів p, d, q дорівнює одиниці, тоді переходимо до п.2.1.

   Якщо визначено першу $ARIMA(p,d,q)(0,0,0)_s$ і знайдено сезонність в лінеаризованих даних за тестом, тоді переходимо до п.2.2.

   Якщо сезонності в початкових даних немає і

   $QS(a_t) > 6$ або $Q(a_t) > \chi^2_{95\%}$, тоді переходимо до п.2.2.

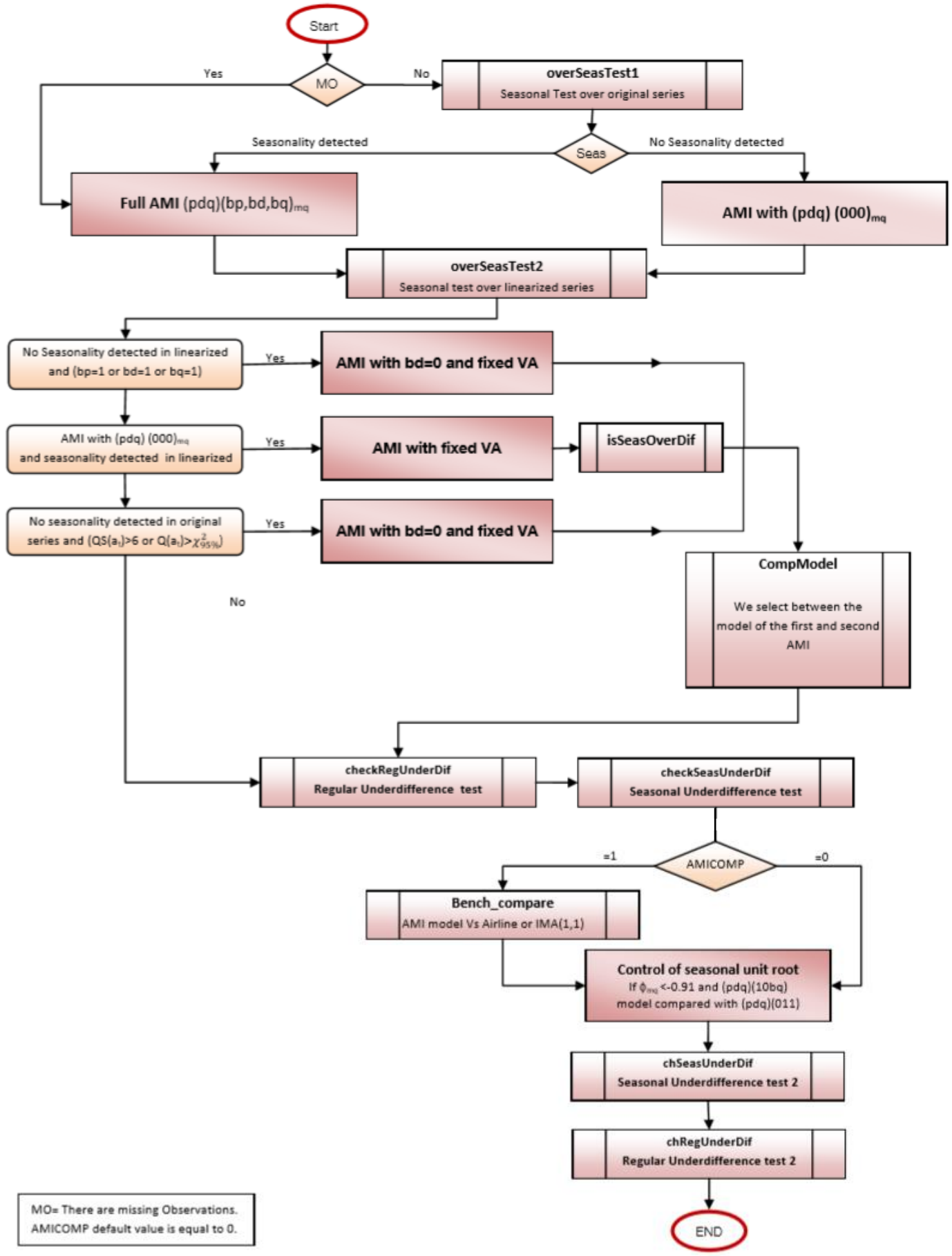
   Якщо жоден з перерахованих вище варіантів не виконується, тоді переходимо до п.5.

   2.1. Визначимо другу ARIMA з сезонним порядком різниці рівним 0 і з фіксованим значенням VA (змінна, яка визначена для встановлення критичного значення). Переходимо до п.3.

   2.2. Визначимо другу ARIMA з фіксованим значенням VA. Переходимо до п.2.2.1.

2.2.1 Виконується перевірка isSeasOverDif. Переходимо до п.3.

3. Вибираємо модель між першою і другою за допомогою CompModel. Переходимо до п.4.
4. Виконується тест chRegUnderDif (ACF test). Переходимо до п.5.
5. Виконується тест chSeasUnderDif . Переходимо до п.6.
6. Перевіряємо встановлене користувачем (або за замовчуванням = 0) значення змінної AMICOMP, яка визначає подальші дії.
   Якщо AMICOMP = 1 і модель не має задовільні результати за тестами, тоді для цієї моделі виконується порівняння Bench_compare з $ARIMA(0,1,1)(0,1,1)_s$ (Airline model) і далі переходимо до п.7.
   В інших випадках (тобто якщо AMICOMP = 0 або AMICOMP = 1, але результати за тестами не є задовільними) поточну модель порівнюють за допомогою Bench_compare з IMA(1, 1). Переходимо до п.7.
   Якщо усе виконується за замовчуванням, тоді без порівнянь поточної моделі переходимо до п.7
7. Виконується контроль сезонної одиниці кореня.
   Якщо $AR(1)AR_s(1)$ є поточною моделлю і один з її коренів перевищує за модулем критичне значення UB1 (за замовчуванням воно дорівнює 0.97), тоді встановлюється значення цього кореню рівне одиниці.
   Якщо $ARMA(1,1)ARMA_s(1,1)$ є поточною моделлю і один з її коренів за модулем перевищує критичне значення UB2 (за замовчуванням воно дорівнює 0.91), тоді встановлюється значення цього кореню рівне одиниці.
   Якщо авторегресійний корінь перевищує значення UBP (за замовчуванням рівне 0.96), то його значення встановлюється рівним одиниці.
   Переходимо до п.8.
8. Виконується тест checkSeasUnderDif. Переходимо до п.9.
9. Виконується тест checkRegUnderDif.
   [5]


Start
MO
Yes
No
overSeasTest1
Seasonal Test over original series
Seas
Seasonality detected
No Seasonality detected
Full AMI (pdq)(bp,bd,bq)mq
AMI with (pdq) (000)mq
overSeasTest2
Seasonal test over linearized series
No Seasonality detected in linearized and (bp=1 or bd=1 or bq=1)
Yes
AMI with bd=0 and fixed VA
AMI with (pdq) (000)mq and seasonality detected in linearized
Yes
AMI with fixed VA
isSeasOverDif
No seasonality detected in original series and (QS(at)>6 or Q(at)>χ²95%)
Yes
AMI with bd=0 and fixed VA
No
CompModel
We select between the model of the first and second AMI
checkRegUnderDif
Regular Underdifference test
checkSeasUnderDif
Seasonal Underdifference test
AMICOMP
=1
=0
Bench_compare
AMI model Vs Airline or IMA(1,1)
Control of seasonal unit root
If ϕmq <-0.91 and (pdq)(10bq) model compared with (pdq)(011)
chSeasUnderDif
Seasonal Underdifference test 2
chRegUnderDif
Regular Underdifference test 2
END
MO= There are missing Observations.
AMICOMP default value is equal to 0.

Puc. 2.1.5

[5]

Більш детальний опис тестів, які згадані в процедурі автоматичної ідентифікації (AMI), записано нижче:

1. overSeasTest1. Вважаємо, що є сезонність, якщо значення QS тесту фіксує сезонність на рівні 99%; або якщо непараметричний критерій сезонності і QS тест зафіксували сезонність на рівні 95%.
2. overSeasTest2. Ми повертаємо параметр OST_{XL}, який дорівнює кількості тестів, які зафіксували сезонність раніше; crQS дорівнює так, якщо QS тест зафіксував сезонність на рівні 99%, в іншому випадку дорівнює ні; crFseas дорівнює так, якщо Fseas тест зафіксував сезонність на 99%, в іншому випадку ні. Ми вирішуємо, що дані сезонні, якщо $OST_{XL} = 1$ або один з crQS чи crFseas дорівнює так.
3. isSeasOverDif . Ця перевірка використовується в тому випадку, коли тест overSeasTest1 вказує на наявність сезонності, генерується перша $ARIMA(p,d,q)(0,0,0)_s$ модель, яку позначимо M, і маємо другу $ARIMA(p,d,q)(0,1,1)_s$ модель, яку позначимо MM, з $\theta_m < 0.7$. Знову виконуємо тест overSeasTest2 на логарифмованому часовому ряді. Якщо виявлено сезонність, тоді CompModel повертає модель MM, інакше – M.
4. CompModel. Це порівняння моделей здійснюється за BIC критерієм, кількістю викидів, кореляцією залишків і кореляцією сезонних залишків. [5] BIC критерій слугує для вибору однієї з декількох статистичних моделей і базується на функції правдоподібності. [1] За замовчуванням, якщо всі статистичні значення, крім BIC критерію погіршуються, тоді модель обирається за BIC критерієм; якщо ж погіршення значення критерію BIC є помірним, а інші статистичні значення покращуються, тоді обирається модель за ними.
5. checkRegUnderDif. Якщо d < 2, p > 0, IMEAN (корекція середнього) = 1 й існує дійсний додатній корінь, який за модулем більше за 0.9, модель порівнюється за CompModel з моделлю $ARIMA(p-1,d+1,q)(P,D,Q)_s$ без середнього. Обирається найкраща з них.

6. checkSeasUnderDif. Якщо $ARIMA(p,d,q)(P,0,Q)_s$ є поточною моделлю і присутня сезонність залишків, тоді мі порівнюємо поточну модель за CompModel з $ARIMA(p,d,q)(0,1,1)_s$, а далі обираємо кращу з них.
7. Bench_compare. Використовуючи порівняння CompModel ми вибираємо між поточною моделлю і:
    - якщо $OST_{XL} > 0$, тоді з моделлю $ARIMA(0,1,1)(0,1,1)_s$ (airline).
    - якщо $OST_{XL} = 0$, тоді з моделлю AMI(1, 1).
8. chSeasUnderDif. Якщо $ARIMA(p,d,q)(P,0,0)_s$ є поточною моделлю і $OST_{XL} \geq 1$ і якщо виконується один з наступних варіантів:
    - присутня сезонна авторегресія
    - присутній ефект робочого дня

    Тоді виконується порівняння за допомогою CompModel з $ARIMA(p,d,q)(P,1,1)_s$ без середнього і обирається краща з них.
9. chRegUnderDif (ACF test). Якщо $ARIMA(p,d,q)(P,D,Q)_s$ є поточною моделлю і виконується один із наступних варіантів:
    - Q-тест відкидається на рівні 95% і в неї є хоча б 17 додатніх коренів авторегресії з 24 перших
    - встановимо мінімальне значення, яке знаходиться між числом 9 і значенням Q з поточної моделі. Встановлене значення позначимо min.

    Якщо перших min лагів є позитивними, тоді порівнюємо за CompModel поточну модель з новою моделлю:
    - якщо d < 2 з $ARIMA(p,d\ +\ 1,min(q+1,3))(P,D,Q)_s$
    - якщо d = 2 з $ARIMA(min(p+1,3),d\ ,q)(P,D,Q)_s$

Потім обирається найкраща з них. Якщо за загальним тестом на сезонність встановлено, що часовий ряд не містить сезонності, тоді мультиплікативні моделі спрощуються на AR(1) і ARMA(1, 1).

Під час деяких тестів поточна модель може бути змінена в залежності результатів тестів. [5]

QS тест. QS тест є варіантом тесту Люнга-Бокса, який розраховується на сезонних лагах, де ми дивимось лише на додатню автокореляцію.
Загальна формула має наступний вигляд:

$$QS = n(n+2)\sum_{i=1}^{k}\frac{[\max(0,\hat{\gamma}_{i\cdot l})]^2}{n-i\cdot l}$$

де k = 2, тому враховуються перші й останні сезонні лаги. Таким чином тест перевіряє кореляцію між фактичним спостереженням та спостереженнями, що були на два роки раніше від фактичного. Зауважимо, що l = 12, коли використовуємо щомісячні спостереження, тому ми розглядаємо автоковаріації між $\hat{\gamma}_{12}$ і $\hat{\gamma}_{24}$ окремо. У випадку квартальних часових рядів визначаємо k = 4. Гіпотеза H_0: дані розподілено незалежно, статистика прямує до розподілу $\chi(k)$.

Значення p-value отримуються за допомогою $P(\chi^2(k) > Q)$ для k = 2. Тому маємо, що $P(\chi^2(2)) > 0.05 = 5.99146$ чи $P(\chi^2(2)) > 0.01 = 9.21034$, $QS > 5.99146$ чи $QS > 9.21034$ пропонує відхилити нульову гіпотезу на рівні 95% чи 99%. [6]

FSeas тест. F-тест на сезонні даммі перевіряє наявність детермінованої сезонності. У тесті використано модель, яка використовує сезонні даммі (середній ефект і 11 сезонних даммі для щомісячних даних, середній ефект і 3 сезонні даммі для квартальних даних) для опису (можливо трансформованої) поведінки часових рядів. Тестова статистика перевіряє, чи сезонні даммі не є разом статистично значущими. Коли цю гіпотезу відкидають, передбачається, що детермінована сезонність присутня.

Цей тест використовує модельну основу χ^2 і F-тест для фіксованих сезонних ефектів, які запропонували Літрас Д. П., Фелдпауш Р.М. і Белл В.Р. (2007), який базується на оцінці регресійних даммі і відповідну t-статистику RegARIMA моделі, в якій частина ARIMA моделі має форму (0, 1, 1)(0, 0, 0). Для щомісячних часових рядів моделі RegARIMA структура полягає в наступному:

$$(1-B)\left(y_t - \beta_1 M_{1,t} - \cdots - \beta_{11} M_{11,t} - \gamma X_t\right) = \mu + (1-B)a_t,$$

де

$$M_{j,t} = \begin{cases} 1 & \text{у місяці } j = 1,\dots,11 \\ -1 & \text{у грудні} \\ 0 & \text{в інших випадках} \end{cases}$$

$M_{j,t}$ є даммі

y_t - початкові значення спостережень у часовому ряді

B – лаговий оператор

X_t – інші регресійні змінні, які використовуються в моделі (викиди, календарні ефекти, фактори, які встановлені користувачем, змінні втручання)

μ – ефект середнього

a_t- змінна білого шуму з середнім значенням рівним нулю і постійною дисперсією

У випадку квартального часового ряду модель оцінюється наступним чином:

$$(1-B)\left(y_t - \beta_1 M_{1,t} - \cdots - \beta_3 M_{3,t} - \gamma X_t\right) = \mu + (1-B)a_t$$

де

$$M_{j,t} = \begin{cases} 1, & \text{у кварталі } j = 1,\dots,3; \\ -1, & \text{у четвертому кварталі;} \\ 0, & \text{в інших випадках;} \end{cases}$$

$M_{j,t}$ у цьому випадку також присутні даммі.

Можна використати індивідуальну t-статистику, щоб оцінити, чи є значущою сезонність для даного місяця, або статистику χ^2, якщо нульовою гіпотезою є те, що всі параметри в сукупності рівні нулю. Тест χ^2статистики є $\hat{\chi}^2 = \hat{\beta}' \left[Var\left(\hat{\beta}\right)^{-1}\right] \hat{\beta}$ у цьому випадку порівнюється критичне значення з розподілом $\chi^2(df)$, з степенем свободи df = 11 (щомісячні дані) або df = 3 (квартальні дані). Так як дисперсія $\hat{\beta}$ обраховується використовуючи оцінку дисперсії α_t може дуже відрізнятися від актуальної дисперсії у невеликих зразках, цей тест коригується за допомогою наступної запропонованої F статистики:

$$F = \frac{\hat{\chi}^2}{s-1} \times \frac{n-d-k}{n-d}$$

де n – це розмір зразку, d – степінь диференціації, s – частота часового ряду (для щомісячних даних рівна 12, для квартальних рівна 4), k – загальна кількість факторів у RegARIMA моделі. Ця статистика відповідає $F_{s-1,n-d-k}$ розподілу за нульовою гіпотезою.

[6]

1.2.12 Оцінка параметрів моделі ARIMA

Оцінка параметрів може відбуватися за наступними алгоритмами:

- Фільтр квадратного кореня (1) [11]
- Алгоритм Морфа, Шидлу і Кайлая, який покращений Мелардом (2) [12]
- Звичайний фільтр Калмана (3) [9]
- Умовна мінімізація квадратів (4) [13]

За допомогою параметру IFILT користувач може визначити спосіб оцінки. В дужках вище наведено значення для параметру і відповідного алгоритму. За замовчування IFILT = 2.

2.2 Прогнозування ціни на акції NVidia за допомогою методу TRAMO

1.2.13 Загальний опис часового ряду з ціною

Розглянемо щоденну ціну акції NVidia з 22 січня 1999 року до 1 січня 2021 року. Зазвичай, дані з ціною на акції подають у форматі OHLCV (Open High Low Close Volume), іноді також включається Adjusted Close, що відображає останню ціну за день після впливової події. Відповідно OHLCV включає ціну під час відкриття ринку (колонка Open), під час закриття ринку (колонка Close), найнижчу за день ціна (колонка Low), найвищу за день ціна (колонка High) і кількість куплених або проданих активів, відображених у базовій валюті (колонка Volume). Отже, я взяла з сайту https://finance.yahoo.com/ OHLCV дані за тікером NVDA (ціна на акції визначено в американський доларах). Відповідно для демонстрації даних із ціною використано бібліотеку yfinance. Вигляд даних зображено на рис. 2.2.1.

```
import yfinance as yf

nvda_data = yf.download('NVDA', start='1999-01-01', end='2021-01-01')
print('Nvidia')
nvda_data.head()
```

```
[*********************100%**********************]  1 of 1 completed
Nvidia
```

Date	Open	High	Low	Close	Adj Close	Volume
1999-01-22	1.750000	1.953125	1.552083	1.640625	1.507891	67867200.0
1999-01-25	1.770833	1.833333	1.640625	1.812500	1.665861	12762000.0
1999-01-26	1.833333	1.869792	1.645833	1.671875	1.536613	8580000.0
1999-01-27	1.677083	1.718750	1.583333	1.666667	1.531826	6109200.0
1999-01-28	1.666667	1.677083	1.651042	1.661458	1.527039	5688000.0

Рис. 2.2.1

Для побудови моделі буде використано середнє значення між найвищою і найнижчою ціною за день. Отже, отримаємо значення середньої ціни і її вигляд зображено на рис. 2.2.2.

```
import yfinance as yf

nvda_data = yf.download('NVDA', start='1999-01-01', end='2021-01-01')
nvda_data['Price'] = (nvda_data['Open'] + nvda_data['Close'])/2
nvda_data.drop(columns=['Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume']).head()
```

```
[*********************100%**********************]  1 of 1 completed
```

Date	Price
1999-01-22	1.695312
1999-01-25	1.791667
1999-01-26	1.752604
1999-01-27	1.671875
1999-01-28	1.664062

Рис. 2.2.2

1.2.14 Визначення параметрів і опис їх запису за допомогою програми JDemetra+

Мета полягає в тому, щоб за допомогою TRAMO побудувати модель ARIMA. TRAMO включає процедуру автоматичної ідентифікації моделі, оцінку параметрів моделі, опрацювання пропущених значень, визначення викидів і деякі тести. Відповідно процедуру TRAMO буде виконано за допомогою програми JDemetra+.

Спочатку завантажимо їх у програму. Відповідно у пункті ‘Providers’ вибравши пункт ‘Spreadshits’ відкриваємо дані з середньою ціною акції. Під час відкриття потрібно вказати період розбиття даних (щомісячні, двомісячні,

квартальні тощо). Визначимо щомісячне розбиття, яке програма робить автоматично. Відповідно графік із щомісячною ціною зображено на рис. 2.2.3:

Графік з ціною на акції NVidia (01.01.1999 – 01.01.2021)

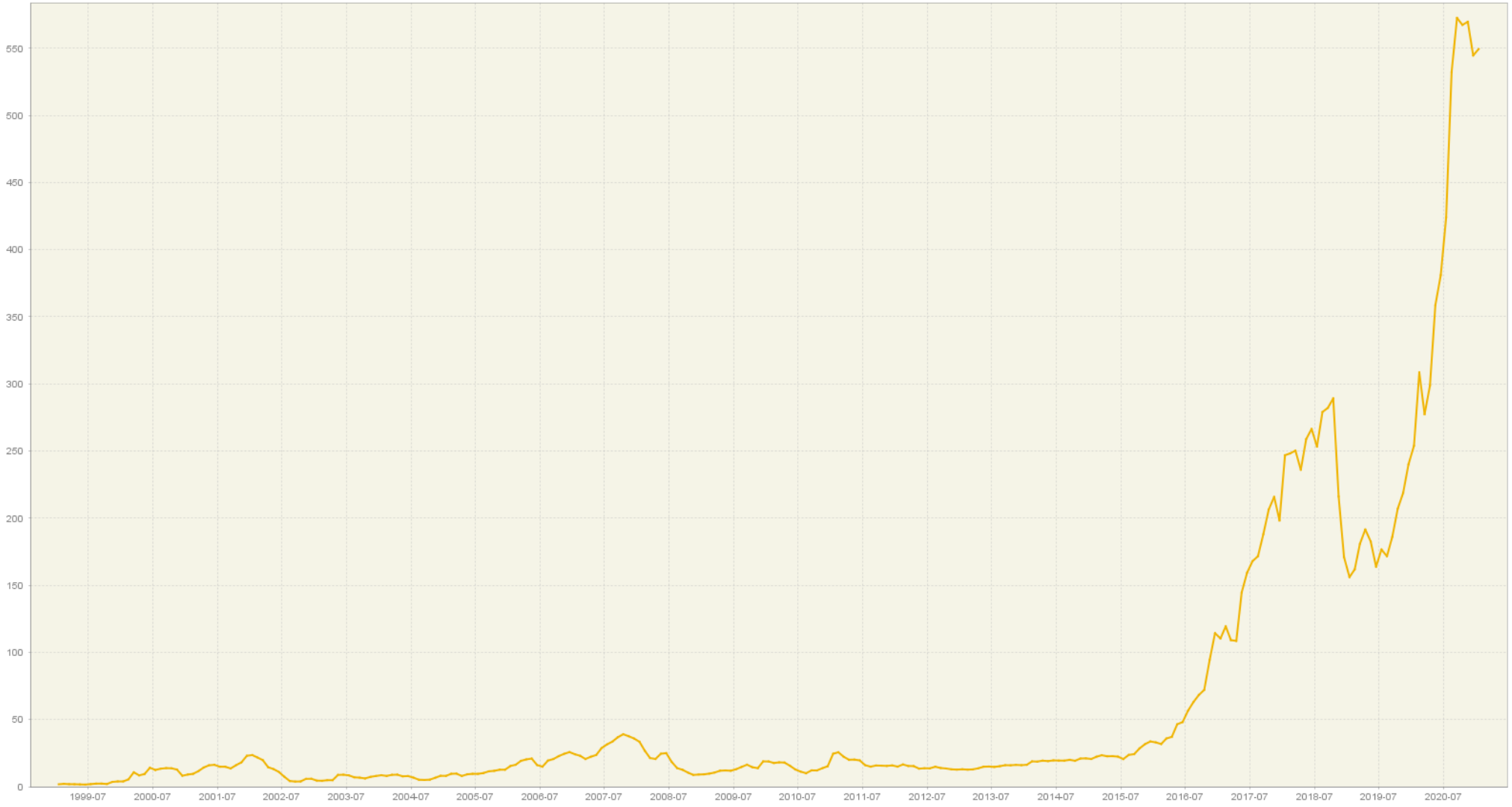


Рис. 2.2.3

Визначимо значень параметрів для методу TRAMO зображено на рис. 2.2.4 (а) і рис. 2.2.4 (б).

⊟ SERIES	
⊟ Series span	All
Type	All
Preliminary Check	☑
⊟ ESTIMATE	
⊟ Model span	All
Type	All
Tolerance	0,0000001
Exact ML	☑
Unit root limit	0,96
⊟ TRANSFORMATION	
Function	Auto
Fct	0,95
⊟ REGRESSION	
⊟ Calendar	
⊞ Trading days	in use
⊞ Easter	in use
Pre-specified outliers	
Intervention variables	
Ramp effects	
User-defined variables	
Fixed regression coe...	

Рис. 2.4 (а)

⊟ OUTLIERS	
Is enabled	☑
Use default critical v...	☑
Critical value	3,5
⊟ Detection span	All
Type	All
Additive	☑
Level shift	☑
Transitory change	☑
Seasonal outlier	☐
EML estimation	☐
TC rate	0,7
⊟ ARIMA	
Automatic	☑
Accept Default	☑
Cancelation limit	0,05
Initial UR (Diff.)	0,97
Final UR (Diff.)	0,91
Arma limit	1
Reduce CV	0,12
LjungBox limit	0,95
Compare to default	☑

Рис. 2.4 (б)

Далі буде описано, що саме задають параметри на рис. 2.2.4 (а) і рис. 2.2.4 (б)

Series/Series span/Type визначає інтервал у часовому ряді, на якому буде застосовано метод Tramo. Визначимо all, бо нам треба застосувати до всього часового ряду.

Якщо позначено *Series/Preliminary Check*, тоді перевіряється якість числового ряду і виключаються дуже проблемні спостереження, тобто спостереження, що повторюються, або спостереження з відсутніми значеннями, які перевищують загальну можливу кількість відсутніх значень (184).

Estimate/Model span/Type визначає інтервал у часовому ряді, на якому буде визначено параметрі regARIMA (якщо маємо регресію з ARIMA).

Estimate/Tolerance визначає допуск збіжності для нелінійної оцінки. Абсолютні зміни в логарифмічній функції правдоподібності порівнюються з цим значенням, щоб перевірити збіжність ітерацій оцінки.

Якщо позначено *Estimate/Exact ML*, тоді виконується точна оцінка максимальної правдоподібності. В якості альтернативи використовується метод безумовних найменших квадратів.

Estimate/Unit root визначає порогове значення останнього тесту одиничного кореня для ідентифікції порядку обчислення різниці. Якщо величина кореню авторегресії (AR) для кінцевої моделі менше цього порового значення, то приймається, що це є одиничний корінь, значення порядку полінома авторегресії (AR) зменшується на одиницю, а відповідний порядок різниці (несезонного, сезонного) збільшується.

Transformation/Function має можливі значення None – відсутність можливих перетворень даних, Log – виконується логарифмування даних, Auto – виконується тест на логарифм/рівень для визначення типу перетворень і потім визначене перетворення застосовується до даних.

Transformation/Fct визначає значення параметру FCT, що визначає чи буде перевага для логарифу, чи для рівня.

Загалом розділ *Regression/Calendar* визначає спосіб введення календарних ефектів у модель (як ефект робочого дня і Великдень).

Regression/Prespicified outliers дозволяє користувачу власноруч ввести спостереження з викидами.

Regression/Intervention variables оцінює наслідки викидів, які введені користувачем.

Regression/Ramp effects включає ефект скату у модель. Ефект скату – це лінійне збільшення або зменшення рівня числового ряду протягом певного інтервалу часу.

Regression/User defined variables дозволяє користувачу включити в модель додаткові змінні (регресори).

Якщо позначено *Outliers/Is enabled*, тоді буде виконано автоматичний пошук викидів на інтервалі, який визначено в *Outliers/Detection span/Type.*

Якщо позначено *Outliers/Use default critical value*, тоді критичне значення визначається автоматично (параметр VA) за допомогою автоматичної процедури пошуку викидів. Якщо не позначено *Outliers/Use default critical value*, тоді

користувачу пропонується самостійно ввести критичне значення в полі *Outliers/Critical value* (за замовчуванням встановлено 3.5).

Далі в пунктах *Outliers: Additive, Level shift, Transitory change, Seasonal outlier* визначається чи потрібно шукати відповідно адитивні викиди (AO), зсув рівня (LS), тимчасова зміна (TC) і сезонні (інноваційні) викиди (IO).

У пункті *Outliers/EML Estimation* визначається за яким методом буде здійснюватися оцінка параметрів на проміжних етапах автоматичного виявлення та корекції викидів. Тобто якщо цей пункт позначено, тоді використовується точний метод оцінки ймовірності, інакше – швидкий метод Ханнана-Ріссанена.

Outliers/TC Rate визначає швидкість спаду для викидів тимчасової зміни (TC).

ARIMA/Automatic визначає, що ідентифікація моделі ARIMA буде відбуватися за допомогою автоматичної процедури ідентифікації. У випадку, коли цей пункт не є позначеним, тоді користувачу пропонується самостійно ввести значення порядків для моделі ARIMA.

ARIMA/Accept default контролює чи може бути встановлена модель за замовчуванням ($ARIMA(0,1,1)(0,1,1)_m$) на першому кроці автоматичної процедури ідентифікації. Якщо Q-статистика для залишків Люнга-Бокса є задовільною, тоді модель за замовчуванням приймається, і подальші кроки ідентифікації не виконуються.

ARIMA/Cancelation limit визначає параметр CANCEL. Якщо модуль різниці коренів авторегресійного (AR) і ковзного середнього (MA) менше, ніж CANCEL, тоді ці два корені скорочуються.

ARIMA/Initial UR (Diff.) визначає параметр UB1, про який згадувалося раніше під час опису автоматичного процесу ідентифікації.

ARIMA/Final UR (Diff.) визначає параметр UB2, про який також згадувалося раніше під час опису автоматичного процесу ідентифікації.

ARIMA/ARMA limit визначає порогове значення для t-статистики коефіцієнтів моделі ARMA. Якщо вищий порядок ARMA має значення t, яке менше за визначене порогове значення, тоді цей порядок зменшується.

ARIMA/Reduce CV визначає відсоток, на який буде зменшено критичне значення для викидів (VA), у випадку, коли визначена модель має незадовільне значення коефіцієнту довіри статистики Люнга-Бокса.

ARIMA/LjungBox limit визначає рівень значущості для Q-тесту Люнга-Бокса, що використовується, коли визначено автоматичну ідентифікацію моделі.

Якщо позначено *ARIMA/Compare to default*, тоді порівнюється модель, визначена за допомогою автоматичної процедури ідентифікації, і модель за замовчуванням ($ARIMA(0,1,1)(0,1,1)_m$). Порівняння відбувається за допомогою діагностики залишків, структури моделі та кількості викидів.

1.2.15 Застосування методу TRAMO з визначеними параметрами до часового ряду з ціною на акцію NVidia

Тепер застосуємо до даного числового ряду метод TRAMO з визначеними параметрами з рис. 2.2.4 (а) і рис. 2.2.4 (б).

Значення порядків і параметрів зображено на рис. 2.2.5.

Arima model

[(0,1,1)(0,0,0)]

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	0,3342	5,66	0,0000

Рис. 2.2.5

Далі розглянемо певні особливості моделі.

З рис. 2.2.6 можна побачити, що вирішується застосувати логарифмування. Також бачимо, що відсутні ефекти робочого дня і Великодня. Також загалом маємо 7 викидів, але про них буде згадано пізніше.

Summary

Estimation span: [1-1999 - 1-2021]
265 observations
Series has been log-transformed
No trading days effects
No easter effect
7 detected outliers

Рис. 2.2.6

На рис. 2.2.7 маємо такі значення для критеріїв AIC, AICC, BIC.

AIC = 1264.3691862338594
AICC = 1265.2387514512507
BIC (corrected for length) = -4.211183819589814

Рис. 2.2.7

На рис. 2.2.8 визначено викиди за типами та їх коефіцієнти, які враховуються в остаточній моделі.

Outliers

	Coefficients	T-Stat	P[|T| > t]
AO (3-2000)	0,4155	6,37	0,0000
TC (5-2003)	0,5212	5,54	0,0000
LS (11-1999)	0,5730	5,39	0,0000
TC (12-2000)	-0,4262	-4,53	0,0000
LS (8-2002)	-0,4487	-4,22	0,0000
TC (1-2011)	0,3829	4,07	0,0001
TC (6-2000)	0,3810	4,03	0,0001

Рис. 2.2.8

Маємо наступний прогноз ціни на рис. 2.2.9:

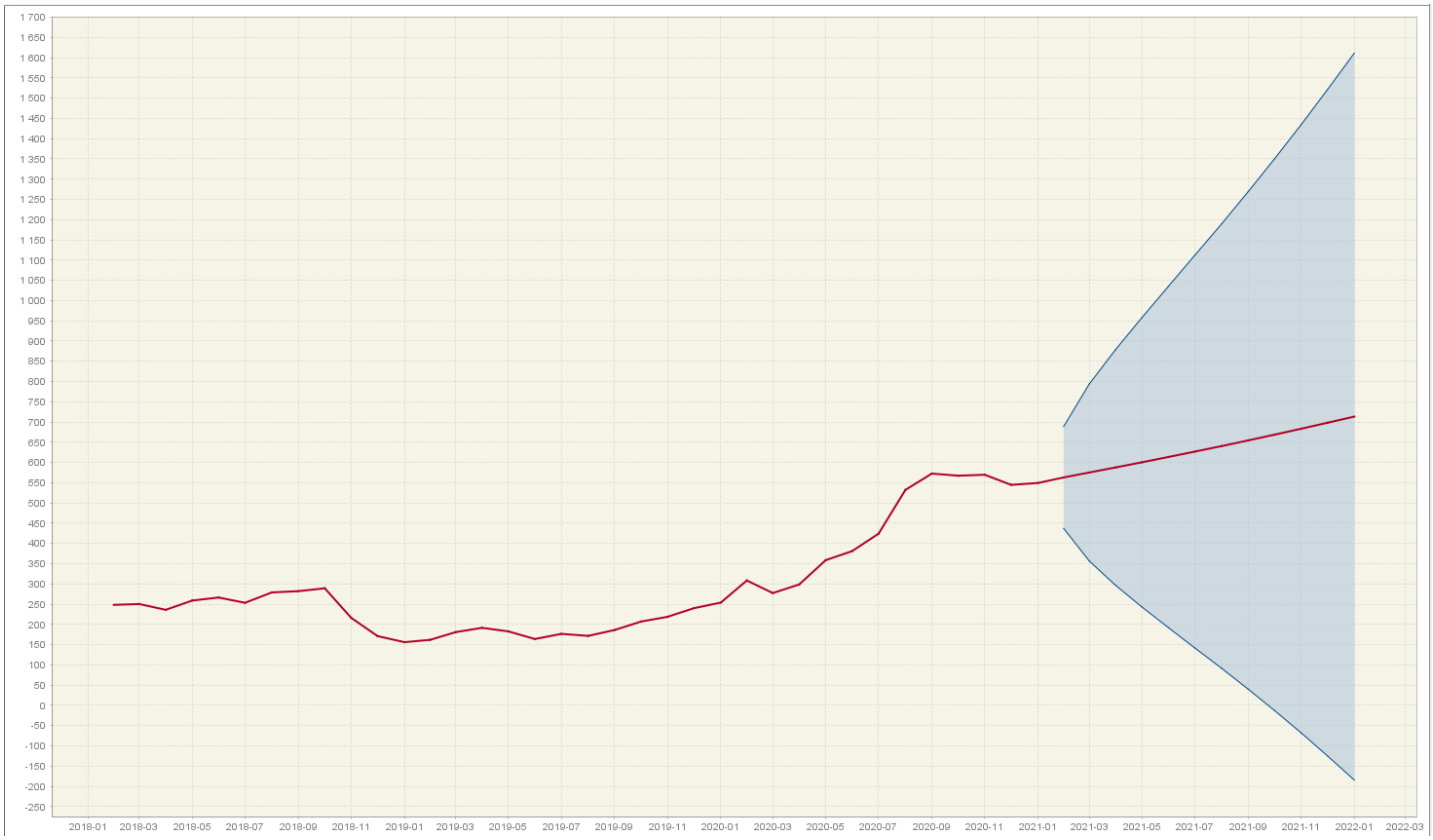


Рис. 2.2.9

Розглянемо прогноз для лютого, березня і квітня 2021 року і порівняємо з реальними значеннями ціни. На рис. 2.2.10 зображено прогнозовані значення і порівняємо ці значення з дійсними, які зображено на рис. 2.2.11.

	y_f
2-2021	562,929
3-2021	575,179
4-2021	587,696

Рис. 2.2.10

Date	Price
01.02.2021	565,505
01.03.2021	509,83
01.04.2021	594,51

Рис. 2.2.11

Можна побачити, що прогнозоване значення на лютий є досить близьким до реального. І загалом прогноз правильно показує тренд ціни на ці місяці.

1.2.16 Застосування методу SEATS до моделі, яку отримано за допомогою TRAMO

Основна ідея методу SEATS полягає в тому, щоб розбити даний часовий ряд на чотири компоненти: Trend-Cycle (циклічно-трендова), Seasonal (сезонна), Irregular (нерегулярна, білий шум), Transitory (залишок). Причому всі ці компоненти є

лінійно незалежні один від одного і подаються у вигляді ARIMA моделі. У сумі вони дають модель, яку отримано за допомогою TRAMO. [7]

Через те, що в нашому випадку в початковій моделі TRAMO немає сезонної частини, тому сезонно скоригована модель буде такою самою. Маємо декомпозицію, яку зображено на рис. 2.2.12.

Model
D: 1,00000 - B
MA: 1,00000 + 0,327916 B

sa
D: 1,00000 - B
MA: 1,00000 + 0,327916 B
trend
D: 1,00000 - B
MA: 1,00000 + B
Innovation variance: **0,44084**
irregular
Innovation variance: **0,11292**

Рис. 2.2.12

1.2.17 Застосування методу X-13 для побудови моделі та її декомпозиції

Метод X-13 є нащадком методу X-11, а саме його покращеною версією. Роль TRAMO в цьому виконує метод reg-ARIMA. Декомпозиція в цьому методі є ітераційною. Під час кожного кроку відбувається оцінка різних компонентів з використанням відповідних ковзних середніх. [6]

Застосуємо метод X-13 до ціни на акції NVidia. Маємо модель, яку зображено на рис. 2.2.13.

Arima model
[(0,1,1)(0,0,0)]

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	0,3756	6,52	0,0000

Рис. 2.2.13

Далі розглянемо чи було вирішено застосувати логарифмування до часового ряду, скільки і яких викидів було знайдено, чи був присутній ефект Великодня і робочого дня.

Summary

Estimation span: [1-1999 - 1-2021]
265 observations
Series has been log-transformed
Series has been corrected for leap year
Trading days effects (1 variable)
Easter [1] detected
5 detected outliers

Рис. 2.2.14

Отже, на рис. 2.2.14 бачимо, що було вирішено застосувати логарифмування, було знайдено п’ять викидів і присутній один ефект Великодня.

На рис. 2.2.15 зображено значення AIC, AICC, BIC критерієв.

AIC = 1282.7263526041381
AICC = 1283.435014021461
BIC (corrected for length) = -4.155321881627507

Рис. 2.2.15

Тепер на рис. 2.2.16 розглянемо, які саме викиди було знайдено й які коефіцієнти вони мають.

Outliers

	Coefficients	T-Stat	P[|T| > t]
LS (3-2000)	0,6761	6,22	0,0000
LS (5-2003)	0,6384	5,87	0,0000
LS (11-1999)	0,5564	5,12	0,0000
LS (1-2011)	0,4557	4,19	0,0000
TC (12-2000)	-0,3951	-4,10	0,0001

Рис. 2.2.16

На рис. 2.2.17 зображено ефект робочого дня.

Working days

	Coefficients	T-Stat	P[|T| > t]
Week days	0,0035	2,84	0,0049

Рис. 2.2.17

На рис 2.2.18 маємо коефіцієнт для ефекту Великодня.

Easter [1]

	Coefficients	T-Stat	P[|T| > t]
Easter [1]	0,0505	2,95	0,0035

Рис. 2.2.18

На рис. 2.2.19 маємо графік прогнозу.

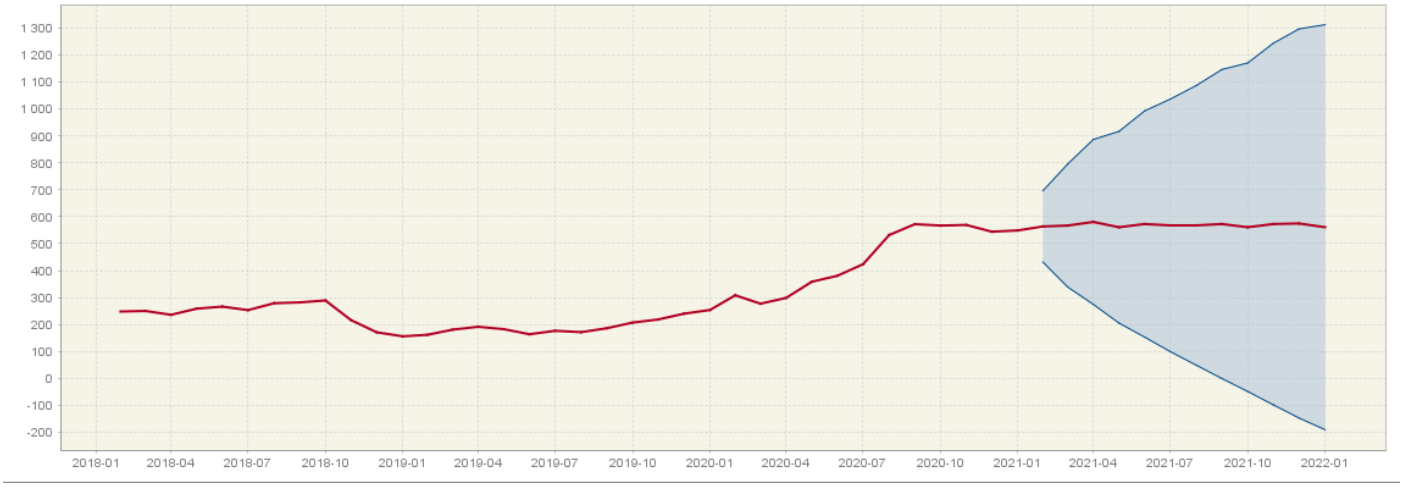


Рис. 2.2.19

Тепер знову розглянемо прогнозоване й реальне значення ціни на акцію. Відповідно на рис. 2.2.20 зображено прогноз, а дійсне значення на рис. 2.2.11.

	y_f
2-2021	564,026
3-2021	567,389
4-2021	580,796

Рис. 2.2.20

Прогнозоване значення на лютий є дуже близьким до реального. Також можемо побачити, що в цілому реальна ціна за ці два місяці збільшилась, так само як і прогнозована.

Тепер розглянемо декомпозицію на рис. 2.2.21. У цьому випадку також сезонно скоригована модель буде тією ж самою, бо немає сезонної компоненти.

Model
D: 1,00000 - B
MA: 1,00000 + 0,334187 B

sa
D: 1,00000 - B
MA: 1,00000 + 0,334187 B
trend
D: 1,00000 - B
MA: 1,00000 + B
Innovation variance: **0,44501**
irregular
Innovation variance: **0,11083**

Рис. 2.2.21

1.2.18 Порівняння результатів методів TRAMO/SEATS і X-13.

Порівнюючи вигляд ARIMA моделей, які зображено на рис. 2.2.5 (для TRAMO/SEATS) і рис. 2.2.13 (для X-13), можна зазначити, що вони мають однакові значення порядків і дуже близькі коефіцієнти. Тому в цілому зрозуміло, що прогнози також будуть схожими.

За TRAMO/SEATS типи всіх викидів, окрім двох (12-2000 і 11-1999), визначено інакше ніж за методом X-13. Також знайдено більше викидів саме за T/S (6-2000, 8-2002).

Також значною відміною є те, що за допомогою X-13 зафіксовано ефект робочого дня і Великодня, а за T/S – ні.

Значення прогнозів є схожими, але за допомогою T/S краще визначено тренд на три місяці, а прогноз на лютий, який отримано за допомогою X-13 є більш точним. Сезонне коригування також є схожим через схожість отриманих моделей.

Значення AIC, AICC, BIC критеріїв є кращим у моделі, яку знайдено за допомогою T/S.

Висновки

У роботі описано з яких частин складається ARIMA модель, визначено коли дана модель буде стаціонарною в залежності від значень порядків. Надано визначення алгоритму автоматичного вибору моделі – TRAMO. Наведено приклад застосування даного методу на часовому ряді з ціною на акцію Nvidia: визначення можливої трансфомації, пошук викидів і визначення їх типів, визначення присутності робочого дня і Великодня, наведено прогноз на наступні 3 місяці (лютий – квітень 2021). Далі наведено застосування алгоритму X-13 і результати його роботи. Порівнюється якість роботи цих методів.

Перелік прийнятих скорочень

AR (AutoReressive) – авторегресія;

MA (Moving Average) – ковзне середнє;

ARMA (AutoRegressive – Moving Average) – авторегресія – ковзного середнього;

ARIMA (AutoRegressive Integrated Moving Average) – інтегрована авторегресія – ковзного середнього;

BIC (Bayesian Information Criterion) – інформаційний критерій Байєса;

AMI (Automatic Model Identification) – автоматична ідентифікація моделі;

IO (Innovational Outlier) – інноваційні викиди;

AO (Additional Outlier) – адитивні викиди;

LS (Level Shift) – зміна рівня;

TC (Transient Change) – тимчасова зміна;

IMA (Integrated Moving Average) – інтегрована модель ковзного середнього;

OHLCV (Open High Low Close Volume) – ціна при відкритті, найвища ціна, найнижча ціна, ціна при закритті, кількість куплених і проданих активів

Використана література

1. George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, Greta M. Ljung, Time Series Analysis: Forecasting and Control, 5th edition (2015) https://libgen.is/book/index.php?md5=BB617EAC4CB4EC545575A49DBD7825DD
2. Rob J Hyndman, George Athanasopoulos, Forecasting: principles and practice, second edition (2018) https://otexts.com/fpp2/
3. Victor Gomez, Agustin Maravall, Program Tramo Program TRAMO "Time Series Regression with Arima Noise, Missing Observations, and Outliers" Instruction for the User, EUI Working Paper ECO No. 94/31 (1994) https://cadmus.eui.eu/handle/1814/510
4. E. J. Hannan and J. Rissanen, Recursive Estimation of Mixed Autoregressive-Moving Average Order, Vol. 69, No. 1 (Apr., 1982), pp. 81-94 (14 pages) https://www.jstor.org/stable/2335856?seq=1
5. Agustín Maravall (Bank of Spain), Gianluca Caporello (Bank of Spain collaborator), Domingo Pérez (INDRA), Roberto López (INDRA) NEW FEATURES AND MODIFICATIONS IN TRAMO-SEATS, December 2014 https://www.cepal.org/sites/default/files/courses/files/01_5_tswnewfeatures.pdf
6. Sylwia Grudkowska, JDemetra+ Reference Manual Version 2.2, Department of Statistic, Warsaw (2017) https://ec.europa.eu/eurostat/cros/system/files/jdemetra_reference_manual_version_2.2_0.docx
7. Guy Melard, One some remarks about SEATS signal extraction, SERIEs (2016) 7:53–98 https://www.researchgate.net/publication/291012669_On_some_remarks_about_SEATS_signal_extraction
8. Юрченко М. Ю., Прогнозування та аналіз часових рядів (2018) http://ir.stu.cn.ua/bitstream/handle/123456789/16992/%D0%9F%D1%80%D0%BE%D0%B3%D0%BD%D0%BE%D0%B7%D1%83%D0%B2.%20%D1%82%D0%B0%20%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7%20%D1%87%D0%B0%D1%81%D0%BE%D0%B2%D0%B8%D1%85%20%D1%80%D1%8F%D0%B4%D1%96%D0%B2.pdf
9. R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems (1960) https://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf
10. IBM Documentation SPSS Modeler Help 18.1.1 https://www.ibm.com/docs/en/spss-modeler/18.1.1?topic=spss-modeler-help
11. B. D. O. Anderson, J. B. Moore, Optimal Filtering (1979) http://users.rsise.anu.edu.au/~john/papers/BOOK/B02.PDF

12. G. Melard, Algorithm AS 197: A Fast Algorithm for the Exact Likelihood of Autoregressive-Moving Average Models (1984) https://dipot.ulb.ac.be/dspace/bitstream/2013/13692/1/AS197.pdf
13. Lawrence A. Klimko, Paul. I. Nelson, On Conditional Least Squares Estimation for Stochastic Processes https://projecteuclid.org/journals/annals-of-statistics/volume-6/issue-3/On-Conditional-Least-Squares-Estimation-for-Stochastic-Processes/10.1214/aos/1176344207.full